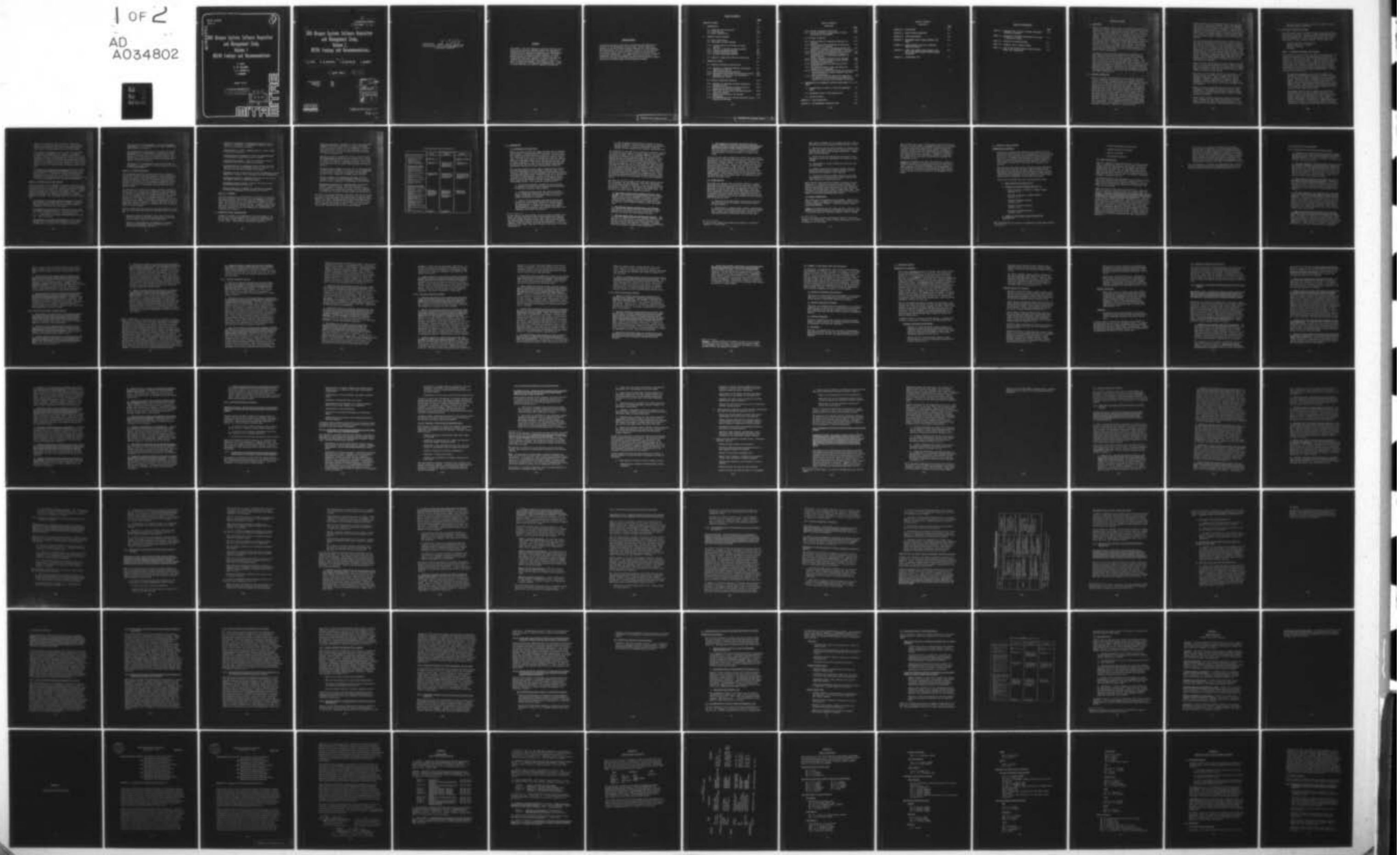
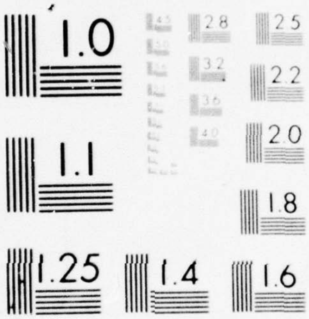
AD-A034 802 MITRE CORP MCLEAN VA F/G 9/2
DOD WEAPON SYSTEMS SOFTWARE ACQUISITION AND MANAGEMENT STUDY. V--ETC(U)
MAY 75 A ASCH, D W KELLIHER, J P LOCHER

UNCLASSIFIED MTR-6908-VOL-1 NL

1 OF 2
AD
A034802




1.0
4.5
5.0
5.6
6.3
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

WASHINGTON OPERATIONS

MTR-6908
Vol. I

1

8.5.

DOD Weapon Systems Software Acquisition and Management Study
Volume I
MITRE Findings and Recommendations

A. ASCH
D. W. KELLIHER
J. P. LOCHER III
T. CONNORS

MAY 1975

MITRE

9

MITRE Technical Report

14 MTR-6908 -Vol-1

Vol. I

6

DOD Weapon Systems Software Acquisition and Management Study. Volume I. MITRE Findings and Recommendations.

10 A. ASCH, D. W. KELLIHER, John Paul ~~J. P.~~ LOCHER III T. CONNORS

11 MAY 1975 12 117p.

CONTRACT SPONSOR	DoD
CONTRACT NO.	None
PROJECT NO.	D40C
DEPT.	D-40

THE MITRE CORPORATION
McLEAN, VIRGINIA 22101

This document was prepared for authorized distribution. It has [redacted] been approved for public release.

MITRE Department
and Project Approval: John Paul Locher III

John Paul Locher III

ABSTRACT

In December, 1974, DoD initiated a two-phase software acquisition study program to identify methods for controlling increasing costs, improving the quality, and minimizing the adverse impact of software in weapon systems. The MITRE Corporation and the Applied Physics Laboratory of Johns Hopkins University were requested to conduct separate, but coordinated, four-month studies in support of the first phase of this study program. This document, Volume I, contains the MITRE study findings and recommendations. Volume II, when published, will provide supporting information.

ACKNOWLEDGEMENT

The MITRE Corporation would like to acknowledge the assistance received from all participants of this study. Special thanks are extended to the DoD Software Steering Committee for their guidance and coordination, and to the personnel at each of the DoD facilities and within industry visited for their cooperation and assistance in providing information. All organizations of the Office, Secretary of Defense, the Military Departments, and industry visited by the MITRE study team were completely cooperative and dedicated to seeking solutions to existing weapon systems software problems.

TABLE OF CONTENTS

TABLE OF CONTENTS

(Continued)

TABLE OF CONTENTS

(Continued)

LIST OF ILLUSTRATIONS

EXECUTIVE SUMMARY

1. BACKGROUND

There is increasing concern about the problems of software cost growth and overruns, schedule delays, and unreliability currently experienced in some weapon systems software efforts. There is also an increasing recognition of the importance of the software roles in the overall mission effectiveness of major DoD weapon systems. These factors led to the Assistant Secretaries of Defense (Installation and Logistics, and Comptroller) and the Director of Defense Research and Engineering issuing a memorandum on 3 December 1974 to establish a joint OSD/Service Software Steering Committee. The charter of the Software Steering Committee is to oversee a two-phase study program to find methods for controlling increasing costs, improving the quality, and minimizing the adverse impact of poor software performance on weapon systems effectiveness.

The MITRE Corporation and the Applied Physics Laboratory of Johns Hopkins University were requested to conduct separate, but coordinated, four-month studies in support of the first phase of the Software Steering Committee study program. This report presents the results of the MITRE study. Volume I contains the findings and recommended high payoff corrective actions, which MITRE recommends be considered for further development and implementation during the second phase of the Software Steering Committee study program. Volume II, when published, will provide the supporting materials and analyses used in the development of the MITRE recommendations.

2. GENERAL OBSERVATIONS

Based on a review of the results of prior studies, the collection of information during the study, and previous MITRE experience, the major contributing factor to weapon systems problems is the lack of discipline and engineering rigor applied to the weapon systems software acquisition activities. This failure frequently leads to over ambitious requirements and subsequent system expansion which causes complex design and redesign problems which then results in delivery delays and poor quality. This deficiency has also resulted in poor documentation, poor test practices and inconsistent review of software progress. The establishment of discipline and engineering rigor includes providing top down control, adherence to various budgets, preparation of specifc documentation, preparation of test plans, use of prototypes, use of independent Verification and Validation capabilities, costed specifications and establishment of meaningful milestones. Additional observations are:

- Sound practices are not being applied to all weapon systems software acquisition efforts. Often, the management practices to provide control and visibility for hardware acquisition are not applied for software. Although good acquisition practices have evolved over the years, the best practices are not always applied to software acquisitions, nor is there an organized method for the exchange of good practices between system management offices and among the Services.

- The current acquisition process does not recognize that the most significant part of a software effort, involving the heaviest expenditures of fiscal and manpower resources, occurs early in the process, before completion of development, in contrast to hardware acquisition where the heaviest expenditures occur during production and deployment. This unawareness, at times, has caused attempts in the acquisition of software to follow the same phases as hardware when, in fact, different acquisition phase definitions are often needed. Also, hardware phasing should take into account uncertainties in the software development effort and relationships with software.

- Total Life Cycle considerations are not adequately covered early in the process of defining software. This oversight has, as an example, caused the late availability of software support facilities and the lack of adequate software maintenance resources for some systems.

- The effect of poor software quality and performance, and delayed software availability on <u>total</u> system costs is frequently much greater than the direct costs for the software. Increased expenditures to improve software development efforts, which would decrease the impact of software on the total system, could result in <u>total</u> system cost savings.

- There is a lack of consistent practices for the feedback of management information on software efforts to allow recognition of successful methods and to identify common, costly problem areas in which attention should be focused for greatest leverage.

- Weapon systems software acquisition problems are similar to the problems that have been identified and, in some cases, resolved for other kinds of system software. Consideration should be given to whether successful practices for other types of software acquisitions apply to weapon systems software acquisition.

3. HIGH PAYOFF CORRECTIVE ACTIONS

Four high payoff areas are defined in which corrective actions would have the greatest leverage on weapon system software costs, quality and timeliness. These corrective actions will support the establishment of discipline and engineering rigor to the acquisition of weapon systems software. The four areas are:

- Software Performance Specification
- Software Acquisition Planning
- Software Technology
- Personnel

AREA I - SOFTWARE PERFORMANCE SPECIFICATION

The corrective actions in this area involve the recognition and consistent application of sound engineering principles and practices to the activities prior to the completion of specifications for software end products. They are intended to provide control over the tendency to be overambitious with functional requirements with inadequate provisions for software development capabilities and under-consideration of other software requirements, such as the provision of capabilities to provide for software maintenance and subsequent modification. The corrective actions provide for:

- The specific documentation which must be prepared in support of the DSARC review process to provide visibility for software and to ensure consideration of the necessary issues. These same types of documents are required to support the review process for weapon systems efforts not under the formal Defense Systems Acquisition Review Council (DSARC) process. Approximately 40% (and possibly greater) of the expenditures for weapon systems software occur outside the formal DSARC review process.

- The methods needed to ensure complete specifications of software end products with consideration given to all factors, such as estimated capacity growth, required maintenance capabilities, allowance for future changes in mission requirements, and provision of facilities for software development and maintenance. These methods would emphasize the orderly, controlled development of mission (functional) requirements considering the cost and schedule impact of each feature that is required. The methods that would provide for definition of all forms of support software and facilities with an early identification of their impact on operational software are also covered.

- The studies and analyses needed to support the definition of software by determining, for example: 1) hardware/ software tradeoffs for meeting mission requirements; 2) the assessment of risks for developing the software which would influence the choice of procurement and management approaches that would be followed to develop software end items; 3) the ability to use software in the DoD inventory to meet requirements for any of the various types of required support software.

- The definition of methods and techniques that may be needed to develop and validate software requirements such as computer models to: 1) assess the operational suitability of defined functional features; 2) determine system sizing parameters; 3) evaluate alternative software architectures; and 4) understand hardware/software interface relationships.

AREA II - SOFTWARE ACQUISITION PLANNING

Corrective actions are recommended to provide a consistent framework and definition of recommended software acquisition management practices for use in planning and conducting the specific software acquisition management efforts for each weapon system. In addition, the factors to be considered and methods to be used in planning and managing the software acquisition for a weapons system are provided. The recommended corrective actions provide for:

- The definition of software acquisition phases, milestones and reporting points applicable to the nature of software development efforts which provide for needed management visibility and control over the process.

- The definition of strategies for handling different types of software acquisition efforts. This effort includes the definition of criteria for use of prototyping and/or parallel development techniques, and procurement practices which would be followed.

- The definition of standard terminology for use throughout the DoD and by DoD Contractors to provide bettern understanding and to facilitate the exchange of information.

- The collection and dissemination of selected management information to provide visibility and assist in the early identification of problems.

- The preparation and maintenance of software and computer inventories to aid the process of determining the availability of existing items and facilities that may have application in new weapon system software efforts, and to identify high payoff areas of software technology effort within or across the Services.

- The factors to be considered by each program manager in developing a specific plan for software acquisition efforts in each weapon system.

AREA III - SOFTWARE TECHNOLOGY

The recommended corrective actions provide for a continued, coordinated Software Technology program to further improve and develop the practices and techniques for software development. Improved technology is needed to establish a sound software development discipline in which roles and terminology are well defined, activities have well-established methods and tools, software status can be determined, costs predicted and reliability assured.

The imposition of discipline and rigor on software acquisition efforts will make it difficult to experiment with improvements in technology. Provisions must be made to provide for real but non-critical programs that can validate and refine the application of new technology prior to being provided to critical programs for use. Much of the advanced technology can be developed best in the context of a specific program rather than in an independent sterile environment. However, care must be taken to ensure that flexible methods are developed which can be applied readily to other programs.

Software technology areas which will have high potential to control the future costs and quality of software include the following:

- Develop quantitative measures of the status of software and its reliability for use in monitoring and predicting progress toward schedule and performance goals.

- Define the characteristics and methods for developing transportable software, capable of being executed in more than one operating environment.

- Develop new approaches for developing software, such as automatic programming, with emphasis on automated aids to prove the correctness of software.

- Develop models to predict software costs at various stages of software acquisition.

- Determine areas and methods for effective standardization of programming languages and support software.

- Conduct pilot programs to apply and consolidate advanced techniques and tools for software development.

- Define principles for selecting computer hardware and software which are mutually supportive and cost effective for meeting functional and performance requirements.

- Determine realistic weapon system software documentation requirements which considers valid development and user requirements.

- Investigate methods for improving effectiveness and reducing real cost of test and evaluation processes.

- Investigate firmware trends and needed DoD policies which provide guidelines for future use.

- Develop techniques and methods for improving the transfer of successful practices across systems and Services.

AREA IV - PERSONNEL

Corrective actions are concerned with the provision of knowledgeable and experienced DoD personnel for the management of software acquisition efforts, and for the design and maintenance of software. The limited scope of this study did not permit the development of definitive recommended actions. A number of the factors that must be considered are provided with the recommendation that this area requires the long-term commitment of OSD and Service attention and resources.

4. CORRECTIVE ACTION IMPLEMENTATION

A number of efforts are recommended for the second phase of the DoD Software Steering Committee study program to implement, and, in some cases, further develop the corrective actions provided in this report. The principal recommended actions are:

- Establish procedures to provide for weapon systems software reviews at OSD and Service levels to support the DSARC review and approval phases for the acquisition of systems. This approach will ensure that early planning is accomplished and that the proper factors have been considered for software acquisition.

- Initiate an effort to collect and analyze selected management information (software cost and progress data) for use by the OSD and Services to measure and control such things as resource utilization, development progress, and for the early identification of weapon systems software acquisition problems.

- Initiate action to update and expand current DoD directives, or issue new instructions, in such areas as the formulation of requirements .acquisition strategies, and management methods for the acquisition of weapon system software.

- Initiate, support, and coordinate technology and study programs for the continued development of improved weapon systems software management and development methodologies.

- Investigate personnel skill classification and selection methods, training programs, and career incentives to develop programs to provide and retain sufficient numbers of management and engineering personnel for the acquisition and maintenance of weapon systems software by the Services.

Section 4 of the report discusses the efforts that will be needed to further investigate and, where appropriate, to initiate implementation of the recommended corrective actions during the second phase of the study. An implementation chart is provided which relates the required efforts to the different corrective actions and to suggested time-phased products. The chart is repeated for general information here in the Executive Summary.

OVERVIEW OF SUGGESTED IMPLEMENTATION OF PHASE II CORRECTIVE ACTIONS

	Possible Products from Phase II Activities		
Corrective Actions (Sections Discussed)	Near Term 2 to 3 Mo.s	Intermediate Term 6 to 9 mo.s	Long Term Greater Than 12 mo.s
● Initiate Interim OSD/Service Policies			
- Require Submission of DCP Supporting Documentation (3.1.4)	● Prepare Letter Directives for OSD Action		● Supercede Letter Directive(s) When Formal DoDD, DoDI Changes Issued.
- Require Collection of Selected Management Information (3.2.6)			
- Review Major Software Documents to Identify Interim Improvements (3.2.7)	● Assign/Allocate Resources; Initiate Review.	● Complete Review of Major DoD Publications Used in Acquisition of Software; Prepare Interim Changes to Cover Major Deficiencies; Initiate OSD/Service Review.	● Supercede Interim Changes When Formal DoDD, DoDI, MIL Standards Issued.
● Initiate Longer Term Actions Leading to Formal Definition of DoD Software Directives, Instructions, MIL Standards			
- Define Major Software Performance Specification Factors (3.1.1)			
- Define Supporting Studies and Analyses (3.1.2)			
- Establish Formal Software QA Practices (3.1.3)	● Prepare Task Statements Assign/Allocate Resources; Initiate Tasks.	● Provide Recommended Policy Changes for Each Task Area Initiated During Phase II for Use as Inputs to Preparation of Formal, Long-Term Policy Improvements.	● Preparation of Draft of Formal, Long-Term Software Acquisition Policy Improvements; Includes Changes to DoDD, DoDI, and MIL Standards and New Publications if Appropriate; Initiate OSD/Service Review.
- Define Major Software Acquisition Planning Factors (3.2.1)			
- Define Software Phases and Milestones (3.2.2)			
- Define Software Acquisition/Procurement Strategies (3.2.3)			
- Define Software Terminology (3.2.4)			
- Define Software Catalogue Requirements (3.2.5)			
● Initiate (and Encourage Existing) High Leverage Studies, Pilot Programs, Research			
- Develop Quantitative Software Engineering Measures (3.3.1)			
- Improve 'Portability' of Software (3.3.2)	● Identify Existing Related DoD/Government/Industry Activities; Prepare Task Statements for New Areas; Assign/Allocate Resources; Initiate New Tasks.	● Establish Methods for Disseminating Research Results and Successful Practices Across Projects and Across Services. ● Disseminate Results from On-Going Research Activities to Services.	● Disseminate Initial Results to Services from Activities/Tasks Initiated During Phase II.
- Investigate Automatic Programming Methods (3.3.3)			
- Define Methods for Predicting Cost of Software (3.3.4)			
- Investigate Areas and Methods for Effective Software Standardization (3.3.5)			
- Establish Pilot Programs Using Latest Methodologies (3.3.6)			
- Define Realistic Documentation Requirements (3.3.7)			
- Investigate Methods for Improving T&E Costs and Performance (3.3.8)			
- Investigate Firmware Trends; Define Policies (3.3.9)			
- Investigate Hardware/Software Architectures (3.3.10)			
- Transfer of Good Software Practices (3.3.11)			
● Personnel			
- Investigate Personnel Practices (3.4)	● Establish Service Review Group	● Submit Recommendations to DoD Management	

1. INTRODUCTION

1.1 Background and Study Goals

Modern day weapon systems are making extensive use of computers and software[1] to perform many combat and other functions which were formerly performed manually, by hardware, or were not able to be performed at all prior to the advent of computer technology. The life cycle cost of acquiring and owning the computer software is becoming significant in relation to the other costs of system acquisition. Also, since the software performs many functions which are critical to overall weapon system mission performance, software is steadily becoming more important.

The importance being placed on weapon system software acquisition and management by the Department of Defense (DoD) is reflected by the number of recent management and technical papers, and committees/panels either sponsored by the DoD or participated in directly by DoD personnel. Several trends in the use of software and stored program computers in the design, development, operation and support of weapon systems are becoming increasingly evident. The trends are characterized by:

1. Growing DoD management awareness of the increasing frequency in the use of and increasing mission dependence on software in military weapon systems.

2. Accompanying suspicion that the costs of software are an increasingly significant portion of DoD costs for weapon systems and that additional indirect costs can often be attributed to software.

3. Concern by DoD management that present methods and controls for acquiring and maintaining software should be improved upon to reduce risks (e.g., cost, schedule, and performance), to improve the software development and maintenance processes, and to improve the quality and timeliness of software end products.

[1] In this report, the term software is used to refer to computer programs, associated data bases, and related documentation required to define, design, develop, produce, test, operate, and maintain the software-related aspects of the total weapon system, including computer hardware, software, personnel and procedures. A list of definitions for common terms used throughout this report is included in Appendix A.

4. Lack of general understanding of how best to impose software management controls without adding inefficiencies, removing incentive or stifling innovation in the fast changing software management and computer technology areas.

In recognition of the need for a focused and coordinated approach for improving weapon systems software management and technical practices throughout DoD, on 3 December 1974 the Assistant Secretaries of Defense (Comptroller and I&L) and the Director, Defense Research and Engineering (DDR&E) established a joint OSD/Service Weapon System Software Steering Committee. Its charter is to identify critical weapon systems management problems and recommend policies and instruments for their solution. In support of the first phase of the Steering Committee activities, The MITRE Corporation and the Applied Physics Laboratory at Johns Hopkins University were requested to conduct separate, but coordinated, four-month studies. Volume I of this report provides the MITRE study findings and recommendations. Volume II provides supporting materials and analyses collected during the study and used in the development of the study recommendations.

The goals for the first phase of the study were briefly defined in a 3 December 1974 OSD memorandum, which also established the DoD Software Steering Committee. A copy of the memorandum is included as Appendix B. These goals are repeated here, with elaboration to indicate the full scope of the MITRE study. In each goal, the effort was to identify and define:

1. The nature of the critical software problems facing the DoD. This required the identification of the critical weapon system software management and software technical problems facing the DoD relative to improving software acquisition and management procedures, to make better use of resources, and to improve software quality and timeliness.

2. The principal factors contributing to the problems. This required the identification of where major software problems are occurring in the weapon system life cycle acquisition process and their causative factors.

3. The high payoff areas and alternatives available. This required the identification of the software areas where OSD and Service attention will have maximum leverage in controlling costs and improving the utilization of software resources, quality and timeliness, and making recommendations for action programs to achieve these improvements.

4. The management instruments and policies that are needed to define and bound the functions, responsibilities and mission areas of weapon systems software management. This has resulted in MITRE-recommended DoD management instruments/policies which are needed to implement an action program to resolve problems in the high payoff areas.

The scope of the study considered all system life cycle phases and all types of software associated with the definition, design, development, test and evaluation, production, operation and maintenance of weapon systems. The term "weapon system" could not be precisely defined. However, the DoD Software Steering Committee provided a list of Army, Navy and Air Force systems for review which tended to bound the study. These reviews excluded intelligence and the ADP (Automatic Data Processing) categories except where ADP software was used in support of a weapon system. They excluded review of the Command and Control and Communications (C^3) systems except for 427M which was included in the review list.

1.2 Study Approach

The study was conducted over a four-month period by a team of MITRE staff from Bedford, Massachusetts and McLean, Virginia. The MITRE Corporation emphasized the software practices of the Department of the Air Force and the Department of the Army, while the Applied Physics Laboratory (APL) emphasized systems of the Department of the Navy and the Department of the Army. Information concerning weapon systems software acquisition and management practices in the DoD was obtained from the following sources during the study:

1. Review of recent DoD software study reports and workshop proceedings and discussions with selected authors of these reports.[1]

2. Preparation of a weapon systems software questionnaire oriented towards identifying major areas of needed software cost, quality, and schedule improvement, and the use of this questionnaire in discussions with Air Force and

[1] A list of these reports and workshop proceedings is included in Appendix C of this report.

Army project personnel on 14 DoD weapon systems.[1] Similar interviews were also conducted between APL and the Navy.

3. Discussions with DoD staff personnel at Service headquarters and command levels who are concerned with establishing DoD weapon system software acquisition, management, and R&D policies.

4. Review of major DoD Regulations and Standards most frequently used in the procurement of software in weapon systems.

5. Interchange of interim findings and ideas with the APL study team.

6. Guidance received from the DoD Software Steering Committee during periodic progress reviews, and from interaction with members of the committee.

7. Soliciting opinions of MITRE technical and management personnel with experience in weapon systems, software acquisition, and DoD practices.

While time and resources did not permit all sources to be examined fully, sufficient correlation existed between data sources to confidently draw conclusions as to the weapon systems software problem areas facing the DoD, their causative factors, and the high payoff areas of needed DoD action. These conclusions and identified high payoff areas were used as the basis for developing recommended DoD actions.

1.3 Report Organization

This study report is presented in two volumes. Volume I contains a summary of findings and recommended corrective actions of direct management interest. Volume II contains further supporting material.

Volume I is organized into four major sections: Section 1 contains introductory information including the purpose and goals of the study; Section 2 contains a summary of major findings

[1] A list of the weapon systems interviewed by MITRE is included in Appendix D of this report. A list of study participants is included in Appendix E of this report.

and concludes with a list of high payoff areas for DoD action; Section 3 presents MITRE's recommended DoD actions in the areas of software performance specification, software acquisition planning, software technology, and personnel; and Section 4 includes a brief outline for implementing the recommended actions during Phase II of the study. Appendices to Volume I are limited only to that information required to understand the content of the four major sections and to initiate the recommended actions.

Volume II is organized into four major sections: Section 1 contains introductory information; Section 2 provides further detail on the 14 systems interviewed; Section 3 contains a brief summation of the major findings and recommendations of the reports and workshops reviewed during the study; and Section 4 includes a software acquisition and management bibliography.

2. SUMMARY OF MAJOR FINDINGS

INTRODUCTION TO SECTION 2

A great deal of documented information was reviewed, and discussions with numerous DoD, Service Headquarters, project office, and private sector personnel were conducted during the course of this study. The weapon system software problems and concerns noted in these documents and expressed as 'lessons learned' by these personnel are voluminous and are generally recorded in the document references for this study and in separate trip reports.[1]

This section of the report extracts from this large information base what MITRE feels are the major weapon systems software problem areas facing the DoD, their causative factors, and the high payoff (high leverage) areas that should be given priority consideration by management in the preparation of future weapon system software acquisition policies and action plans.

This section is divided into the 3 following topic areas:

1. Major Observations (Section 2.1)

2. Discussion of Study Findings (Section 2.2)

   - Characteristics of Software in Weapon Systems (Section 2.2.1)

   - The Cost of Software in Weapon Systems (Section 2.2.2)

   - Software Management Methods (Section 2.2.3)

   - Software Acquisition Methods (Section 2.2.4)

   - Software Development Methods (Section 2.2.5)

3. Summary of High Payoff Areas for DoD Action (Section 2.3)

[1] The references and trip reports are summarized as supporting material in Volume II.

- Software Performance Specification
- Software Acquisition Planning
- Software Technology
- DoD Personnel Practices

2.1 Major Observations

There is a tendency in this type of study to develop sizable lists of problems and causative factors, most of which are already known and agreed upon by the military and civilian organizations most involved. To some degree, this report exhibits this tendency. However, some redefinition of the major problems was necessary to assist in understanding their scope and importance, and in order to formulate a meaningful program of corrective actions.

An attendant danger is the possibility of not clearly identifying to the reader the basic and major causative factors -- of which many problem areas are really only symptoms. One such major factor is sufficiently important to be discussed separately in this section.

Based on a review of the results of prior studies, the collection of information during the study, and previous MITRE experience, the major contributing factor to weapon system software problems is the lack of discipline and engineering rigor applied consistently to the software acquisition activities. This factor is not unique to weapon systems but is symptomatic of software problems in many DoD system areas. This lack of discipline has often resulted in poor design and software requirements control; poor quality and utility of software end products and support functions; inefficient test and validation; and inconsistent review and management emphasis on software progress and system impact. No one action will provide this discipline across all areas of the software acquisition process. Rather, corrective actions must be initiated in many areas. The establishment of discipline and software engineering rigor must include provisions for:

> Total life cycle planning; top down design controls; establishment and adherence to software budgets; definition and use of common, quantitative software performance measures in software development activities; use of acquisition and procurement strategies, milestones, and decision points specific for software; adherence to formal software quality assurance methods; development of efficient verification and validation capabilities; and management controls (checks) to ensure that these disciplines are applied.

The establishment of discipline and engineering rigor forms the basis for the high payoff areas identified in Section 2.3 and for the program of corrective actions developed in Section 3.

2.2 Discussion of Study Findings

2.2.1 Characteristics of Software in Weapon Systems

1. Significant differences exist in the types of weapon systems and in the types and characteristics of software in weapon systems. For example, software in airborne systems differs considerably from software in ground based tactical information systems.

2. The major elements of weapon systems software are often not integral (imbedded) with the operational components, but rather are in the subsystems required to develop and support them (e.g., automatic test equipment (ATE), integration and Validation and Verification (V&V) facilities and ADP-type support). This observation is particularly true for avionics and missiles systems -- less true for tactical control and information systems.

3. Weapon systems software by its nature does not fit previously defined procurement categories. Software is not exactly data nor physical property such as hardware. Attempts to define 'software' in existing terms often causes confusion and often subjects it to inappropriate regulations by those required to manage weapon systems software.

4. Even when software is not a primary cost or delivery item, it can often have large impact on system cost and schedules. Software planning emphasis should be proportional to its importance in the system, rather than the level of the software in the system or its relative cost. Delays in delivery of software or poor quality software can have a very large impact on the total system cost, performance and availability schedules. These indirect costs are often a more significant factor than the direct software acquisition costs.

5. Weapon systems mission requirements are constantly changing and should be viewed as evolutionary by management. The nature of software (e.g., flexibility, relative ease of changeability, and ease of field retrofit) often encourages the use of software solutions to effect mission changes and often to correct for deficiencies in other subsystem areas. While this use of software is believed to represent cost-effective system solutions, management

should be aware of the associated software cost, performance, schedule impacts, and of the need to develop and update software resources, tools, and maintenance facilities.

6. At the present time, weapon systems software is not generally transportable between projects ('portability'). Some exchange of software is being accomplished between projects in the ADP support areas (e.g., programming languages, compilers and utilities). However, for the 14 systems interviewed, no cases were noted where there was an exchange of application software programs. Attempts to enforce rigid standardization of all software may prove counterproductive and should be approached cautiously.

7. The reliability of weapon systems software has become an important issue because of its impact on the overall mission effectiveness of weapon systems. In essence, there may be only one opportunity for it to work properly. This constraint calls for a greater degree of concern that the software will perform as required, when needed, than exists for ADP software, for example. This characteristic of weapon systems software must be considered when formulating and implementing corrective actions.

2.2.2 The Cost of Software in Weapon Systems

1. The indirect total system costs are frequently greater than direct software costs due to poor software quality and performance, and delayed availability in many cases. A total cost savings could result from increased expenditures on software efforts which would reduce the adverse impacts of software on the total system.

2. The total cost of weapon systems software to DoD is apparently increasing, and the ratio of software to computer hardware costs is also increasing. This increase is because of more computers and software being used, and to generally decreasing computer hardware costs.

3. Software differs from hardware in that major costs are generally incurred in development and in operation and maintenance (O&M) phases, not in the production/deployment phase. This fact should be recognized in planning for allocation of fiscal resources.

4. Quantitative software cost information was requested for each weapon system reviewed, but meaningful information was not generally available.[1] This was apparently due to lack of common definitions for the components of software costs, to regulations not requiring software to be broken out and maintained separately from hardware, and to lack of detailed historical cost records. It was also noted that cost information was rarely correlated with technical information for management purposes. Some exceptions were found. For example, in one instance, a detailed software cost breakdown structure and reporting procedures had been initiated. In other instances, portions of annual software contract costs were available but not total costs and related overhead and facility costs.

5. Certain causative factors were found to be frequent contributors to software cost and schedule growth in weapon systems. These factors were identified in past DoD studies as well as during the MITRE study interviews. They include: a) poorly formulated initial software requirements; b) changing requirements and requirements growth during the development phases; c) false starts and need to educate involved organizations before useful output was obtained; d) inefficient use (proliferation) of already existing resources; e) inefficient testing and verification tools and methods; and f) improper use (poor tailoring) of standards and guidance documents in specific procurements.

[1] Note: Future efforts to determine the cost of software in weapon systems should include (start with) the development of a management cost model and agreement on its content. The model should define the software cost components and identify which defense systems, personnel, and facilities are applicable. Once such a model is agreed upon, more meaningful data can be collected and total cost estimates derived. In Appendix F of this report, a possible management weapon systems software cost model is presented. Time and the limited scope of this study did not allow for the collection of sufficient data points to arrive at a defensible cost estimate. However, at the request of the study sponsor, we have provided a gross estimate which is developed in Appendix F. This estimate places direct weapon systems software costs at $.8 to $1.6 billion annually. The room for error and misinterpretation is very large in the approach used, and until such time as the model can be widely reviewed, redefined, and more accurate cost data points obtained from actual system managers, the estimate should be viewed and used cautiously.

6. Formal definition, reporting, collection, analysis and feedback of weapon systems software cost information would improve management's visibility of software. It would provide information in the future so that major areas could be identified where DoD software costs are occurring and thus identify areas for possible improvements in cost and performance.

2.2.3 Software Management Methods

1. Many of the problems identified by previous studies stem from procurements which were started several years ago when the management of software for weapon systems was relatively new to the Services. The Services are generally applying "lessons learned" to recent procurements, and thus some improvements can be expected in cost, schedule, performance and maintainability of weapon systems software.

2. The Services have started organizational, technological, and management programs for the improvement of weapon systems software acquisition management. They have established organizations at command headquarters and within system program management offices whose primary responsibilities involve the management of weapon systems software acquisition. The Services have initiated technological programs which are more oriented toward the specific needs of weapon systems software acquisition and maintenance, as opposed to the broader categories of automatic data processing software. They are also developing guidance documents for use by program managers for the acquisition management of software. Some of these management improvement programs will realize early returns, but other improvements will require time for confirmation of research results against real military problems of software acquisition.

3. Many of the software acquisition and management problems can be traced to inadequate requirements formulation and the need for more detailed planning during the early stages of weapon system acquisition. Examples are the lack of adequate maintenance capabilities for field personnel use, or inflexible software designs that cannot readily adapt to changing mission requirements, or redundant efforts to develop identical types of support software. Further, many of the systems experienced changing requirements as the software was being developed. More adequate planning for software projects is only beginning to be emphasized. Requirements for conscious software acquisition strategies,

suitable development and maintenance support, and realistic intermediate milestones are beginning to be recognized at program office levels. Formal Computer Program Development Plans are being used in some projects. (For example, a "Computer Resources Integrated Support Plan" to assure software maintenance support has been defined by the Air Force.) Nevertheless, software planning is often slighted. Software portions of weapon systems are not always separately identified in contracts, support facilities for software are not specifically provided for, and plans are often not developed to define which software components shall be built and delivered and in what order.

4. Weapon systems software does not have the same degree of visibility, attention and controls as hardware. Software acquisition managers often report a lack of "visibility" for software, by which they mean the ability of someone removed from the actual development to know just how well a software development is progressing. This lack may reflect the absence of measures to monitor software progress and status. It also may reflect an attention to software appropriate to its generally low proportional cost in a system. The importance of software in a weapon system is generally greater than its direct cost since it can have a large impact on system schedule or performance.

5. Little historical cost, schedule, and performance data are available on software development experience which can be used to validate good or bad acquisition practices and to guide development of future acquisition management policies. Software management information is generally not maintained nor available, and thus techniques and methods for its use for making predictions and for improving management policies have not been developed.

6. DoD managers at all levels often use unrealistic assumptions about the capabilities of software, the resources and time required to develop it, and the likely characteristics of a delivered software package. Software rarely works the first time and requires special tools and facilities to develop and validate. It is generally difficult and costly to modify. Software development frequently requires design, test, and redesign iterations before it is satisfactory. Even then, delivered software does not usually perform as expected, both because of the undiscovered bugs which are exposed only in operational use and because of the need to change software to meet

changing or misinterpreted operational requirements. All of these statements are generally agreed upon by practitioners but are often not reflected in management assumptions used as a basis for schedules, cost estimates, and resource allocation.

7. A high turnover rate of military software management personnel was noted in almost all program offices visited. We were impressed by the quality of software management personnel in the various weapon system project offices. However, a very high turnover rate of these same personnel was noted (both leaving Government and the software career areas). There is a continued need to provide training methods to develop new and capable software managers in Government, and career incentives to retain them.

2.2.4 Software Acquisition Methods

1. Although some policies and standards have been established for software acquisition, there does not exist a common set of practices and disciplines in common use. This differs from hardware acquisition where DoD procurement regulations and standards have generally evolved for the management of hardware and total systems.

2. Many of the management principles for hardware acquisition that provide visibility and control are applicable to software acquisition, although in practice these principles have not been generally followed for software. However, it is also important to recognize that there are differences between hardware and software acquisition efforts. For example, production and maintenance have different meanings in the software community than they have for hardware. Software development often begins late in the system development process and is completed early, with continuing modifications. The steps or stages of software development correlate poorly with the acquisition phases defined in DoDD 5000.1. These differences imply a need for allocating resources differently for software projects or software portions of a system program then for hardware.

3. Software acquisitions suffer from many of the same problems as hardware or system acquisitions. For hardware and system acquisitions, overruns and over optimistic estimates of cost and performance (resulting in overruns) have been blamed on unclear or unstable requirements, buy-ins, a rapidly changing technology and a complex contracting structure. The same factors influence software acquisitions,

perhaps to a greater extent than hardware acquisitions. Software requirements are less formally stated, software technology changes even more rapidly than hardware technology in general, and effective procedures for software contracting are still being developed.

4. Many of the software problems are common to all systems; other problems only to specific systems or circumstances. There is no single cure for all problems. Many solutions must be multi-faceted, with improvements needed in both acquisition management and technology areas.

5. While emphasis was placed on weapon systems software, many of the problems observed for this area were the same problems identified in software studies for other kinds of DoD systems, and of large, complex software systems in general. Part of the reason for these common problems is that a large part of the software needed by weapon systems is very similar in character to the types of software used in other systems. Therefore, when implementing corrective actions for weapon systems software, consideration should be given to the applicability of successful practices found for the management and development of software for other types of systems.

6. Problems with acquisition, with the nature of software, and with the lack of engineering discipline make software development inherently risky. Known risk reduction techniques need to be employed for software. Software developments which appeared to be most orderly are those based on previous similar software and continuity with a single contractor(s). Examples are the Minuteman III, Safeguard, and TSQ-73. Where software developments are new or represent significant departures from previous work, or involve new contractors, they should be assumed to require risk management methods. Few cases were noted where software is developed using acquisition strategies intended to reduce risks such as through parallel development, software prototyping, or software feasibility demonstrations.

7. The OSD DCP/DSARC review process for major systems is generally keyed to systems in development phases and to total dollar thresholds. This process often bypasses major software subsystems because either they are in a major software redesign/update phase, but past the equivalent of the

DSARC III decision point, or the software is in a 1-or 2-of-a-kind system (with a relatively low production cost). However, it was noted that major software subsystems are reviewed by the Services, and during the budgetary review processes.

8. There is an ongoing need to ensure that tactical systems interface properly under combined Service operations. Interoperability problems often involve software solutions and could represent major software cost impacts unless interfaces are rigidly controlled in the future. No single unified tactical user group with the long term authority and mission role to ensure interoperability across all tactical systems was noted by MITRE during the interviews.

2.2.5 Software Development Methods

1. There is a general need for better definitions of software terms, measures of software qualities, and the methods of measuring them. For example -- software, software costs, software status (e.g., progress milestones), and software 'quality' (e.g., reliability, maintainability, portability, productivity) -- do not have generally accepted definitions, measures, or methods of measurement in government or in industry.

2. Software technological improvements particularly aimed at developing a software engineering discipline are being made by industry, academia and the Services but require application to real military systems (in addition to laboratory or experimental systems) for evaluation and confirmation. Software technological areas in which research is being conducted with potential application to reducing the costs and improving the quality of military systems include requirements formalization, software development and testing tools, automatic programming, and improved methods for design and implementation (e.g., structured programming).

3. Few military mechanisms exist for transferring proven technology to acquisition programs or for sharing successful practices across acquisition programs and across Services.

4. Several new software technologies are developing which require DoD guidelines for their use in weapon systems. The technologies include: (1) the use of distributed computer processing capabilities among networks of small computers; (2) the use of microprocessors and firmware[1] which incorporate increasing amounts of the computer programs within a system; and (3) the use of on-line interactive programming to an anonymous computer for supporting software development. MITRE found no specific guidelines or instances of common practices for employing these developing trends in weapon systems.

[1] Firmware - Firmware differs from software in that it is not easily alterable such as for software. Use of read-only memories (ROMs) or programmable read-only memories (PROMs) in processors or special purpose hardware are examples of firmware.

2.3 Summary of High Payoff Areas for DoD Action

In developing a recommended DoD course of action for improving the management and control over costs, the quality, and the timeliness of software in weapon systems, it is necessary to first extract from the study findings the areas of highest payoff -- that is, those areas where corrective DoD actions will exert the highest leverage. The following is a brief discussion of the four major areas which are felt would have the greatest leverage and which deserve special OSD and Service attention. Corrective actions in these areas will support the establishment and application of discipline and engineering rigor to the acquisition of weapon systems software. They form the basis and organization of the actions recommended in Section 3 of this report.

1. Software Performance Specification

This area is concerned with the establishment and consistent application of sound engineering principles and practices to the process of specifying software end products.

2. Software Acquisition Planning

This area is concerned with the establishment of a consistent framework and the definition of recommended software acquisition management practices that should be used in planning and conducting weapon systems software acquisition management efforts.

3. Software Technology

This area identifies specific software technology programs needed to further improve and establish software development and management practices and techniques.

4. Personnel

This area is concerned with the provision of knowledgeable and experienced DoD personnel for the management of software acquisition efforts, and for the design and maintenance of software.

3. CORRECTIVE ACTIONS

INTRODUCTION TO SECTION 3

This section recommends specific corrective actions which should be considered during Phase II of the study. The actions are ordered by the four high payoff areas previously discussed in Section 2.3. The emphasis of the actions chosen is to stress the early establishment of a discipline and a software engineering rigor which needs to be applied to the weapon system software acquisition process. The approach chosen recommends that OSD and the Services develop consistent DoD-wide software guidelines which provide for more comprehensive planning and expenditure of software-related resources early in the development process in order to improve the overall life cycle costs of software and weapon systems, and to improve the quality and timeliness of software end products. The intent is not to provide a 'cookbook' approach for the acquisition of software but, rather, to provide a set of proven software guidelines which can be tailored for the specific weapon system under consideration, and the necessary management controls for their use.

Much of the material presented in support of the actions is to provide the reader with a flavor for the type and level of detail required and is not necessarily complete in itself. Rather, the material is presented to identify and bound the nature of the activities that are recommended be pursued during Phase II of the study. In most instances, much more extensive material on any one subject can be obtained from a number of Service and industry publications.

A summary listing of the major actions follows. A detailed discussion of each action is presented in the indicated section.

Software Performance Specification

- Identify the important weapon systems software requirements and performance specification factors and establish specific DoD guidelines to ensure that these factors are adequately considered in the software development process (Section 3.1.1).

- Require specific software design tradeoff studies and analyses as part of the performance specification process (Section 3.1.2).

- Establish formal software Quality Assurance (QA) practices which require the use of proven software design, development, and validation methods (Section 3.1.3).

- For 'major' systems, require specific software supporting documentation and analyses as part of the OSD level DCP/DSARC review process. Require similar supporting documentation and analyses at Service levels for 'non-major' systems or systems past the equivalent DSARC III decision point but involved in a major update cycle (Section 3.1.4).

Software Acquisition Planning

- Identify the important weapon systems software acquisition and life cycle planning factors and establish specific DoD guidelines to ensure that these factors are adequately considered in the software acquisition planning and management process (Section 3.2.1).

- Define specific acquisition phases and milestones for weapon systems software which reflect the true nature of software development in the overall system acquisition process. Define related guidelines for use by project planning personnel (Section 3.2.2).

- Define specific weapon systems software acquisition and procurement strategies (such as software prototyping and parallel development) which maintain contractor incentives and limit software development risks. Define related guidelines for use by project planning personnel (Section 3.2.3).

- Establish common definitions for software terminology for use throughout the DoD and by DoD contractors (Section 3.2.4).

- Establish methods for identifying DoD resources applicable for use across systems and Services; for example, through the preparation and maintenance of a DoD catalogue (inventory) of weapon systems computer hardware, software and facility resources. Define related guidelines for use by Service level and project personnel (Section 3.2.5).

- Initiate OSD action to require the collection and dissemination of selected weapon systems management information including software-related cost, technical performance, and schedule information (Section 3.2.6).

- Review major DoD publications (i.e., directives, instructions, regulations, and MIL standards) used in the acquisition of software in weapon systems. Initiate interim changes to correct for software omissions, deficiencies, and conflicts until formal long-term solutions are implemented (Section 3.2.7).

Software Technology

- Ensure that research, studies, and pilot programs are initiated or continued in areas where current technology and management practices are inadequate in meeting the requirements for efficient development of reliable software and for effective management control of the development process. Eleven areas are discussed which should be given high priority in DoD allocations of software R&D funds (Section 3.3).

Personnel

- Investigate and establish methods for improving software personnel selection and training practices and for developing personnel incentives (Section 3.4).

A certain amount of overlap and redundancy was necessary in developing the material in the following sections. For example, the preparation of the software acquisition planning material includes some performance specification considerations. However, this redundancy was felt necessary in order to provide DoD with a comprehensive discussion in each subject area.

3.1 Software Performance Specification

The corrective actions in this area concern the recognition and consistent application of sound engineering principles and practices to the process of specifying and validating the requirements of software end products. They are intended to provide for control over the tendency to overspecify the functional requirements with the attendant risk of under-specifying other software requirements, such as the provision of capabilities to provide for software maintenance and subsequent modification.

3.1.1 Checklist of Important Software Performance Specification Factors

Recommended Action: Identify the important weapon systems software requirements and performance specification factors and establish specific DoD guidelines to ensure that these factors are adequately considered in the software development process.

While there are significant differences in the types of weapon systems and in the nature and complexity of the software required to develop, operate, and support them, certain common factors exist which should be recognized early in the requirements definition and performance specification and validation process. Many of these factors are addressed in Service level publications and handbooks, but are not currently being consistently applied across all systems. This section lists several important factors which should be formalized during Phase II of the study. It is not intended to be a complete list but rather to indicate the nature of the required DoD guidelines. Several items are further expanded in later sections.

1. Recognize total software life cycle requirements. The requirements definition and performance specification process should apply to all software end items including operational software, support software (e.g., software development tools, test and validation software, and operations and maintenance support software), automatic test equipment and diagnostic software, and training/simulation software. Special emphasis should be applied to ensure that software maintenance requirements are considered.

   The organization of the software requirements should consider the following categories: mission requirements needed to support the overall system mission; operations and maintenance requirements needed to support and maintain the system after transition; system design requirements needed to ensure that the software capabilities and performance are compatible with total system requirements; and software development requirements needed to ensure that all resources and facilities required to develop and validate the software are considered.

2. Approach the development in an orderly fashion. Require an overall approach and strategy for specifying, developing, and validating the software. For example, understand when each software component is required, who will be responsible for developing and validating it, and the risks and dependencies involved.

3. Establish strict controls over software functional and performance (mission) requirements during the program. To protect against software-related cost and schedule growth and computer hardware and software performance degradation caused by uncontrolled changes and user requirements growth, a system for prioritizing software requirements in major defense systems should be established. For example, at the time of the initial software life cycle planning and requirements definition, software requirements should be identified as either high priority (essential to mission success), medium priority (necessary for most effective operation), or low priority (aids, or nice features, but not necessary for system operation). The priorities should have concurrence from the user command and be used to delete requirements when necessary to control cost, performance, and schedules during the program.

4. Evaluate the use of software versus hardware or other design approaches. To ensure the use of software design approaches in weapon systems only when software represents the most beneficial design choice, a separate analysis of software versus other design approaches (including hardware, firmware, or manual procedures) should be performed during the initial validation phase.

5. Choose a software architecture which best reflects the weapon system requirements. The software design approach chosen should consider all weapon system requirements including reliability, maintainability, modularity, future growth, hardware capacities and capabilities, interfaces and interoperability with other systems.

6. Evaluate use of new software developments and facilities versus use of existing resources. To ensure the efficient utilization of existing DoD computer hardware, software, and facilities before initiating new developments, an analysis should be performed during the initial validation phase. The analysis should consider use of existing computer hardware designs and software (including operating systems, application programs, support software; i.e., utility programs, languages, compilers, assemblers, testware, maintenance tools), and possible shared use of existing software maintenance and validation facilities.

7. Establish software related performance standards and software sizing budgets (set quantitative goals for software performance) as well as functional requirements. Quantitative software performance standards (e.g., response time for operator (user) inputs under a stated processing load) and software sizing budgets (e.g., estimates of the number of words of code and execution times at a subroutine level) should be established apart from the mission performance requirements and used as a management tool during the development phases.

8. Recognize software development dependencies. Recognize software development dependencies such as the need for development tools (e.g., compilers, assemblers, utilities) and computer facilities before the start of coding. Take these dependencies into consideration during the contractor selection process and in the overall development planning. Special emphasis should be applied to ensure that proprietary and ownership rights of development and maintenance tools and facilities are considered.

9. Chose a performance specification approach which allows for a phase-in period where a new contractor is involved or the user requirements have not been previously verified. Don't expect the contractor to be an expert in the user requirements, nor the user to know the details of his own requirements without a trial demonstration and evaluation iteration.

10. Design-in sufficient system expansion and modularity capabilities. Assume that software requirements will grow during development and after system transition, and that additional resources (e.g., storage, compute time) will eventually be required.

11. Emphasize ease of change in the software performance specification process. Recognize that weapon systems software requirements will change over the life of the system. Where appropriate, consider use of a modular architecture which allows for changing application program requirements.

12. Control the introduction of software changes during the development process. One of the most difficult and costly problems is the tendency for the user to add or change requirements while the developer is attempting to design, code and debug his software. Consideration should be given to use of an early design freeze on requirements with the incorporation of valid changes introduced as packaged changes later in the process.

13. Define explicit interface requirements for external interfaces as early as possible. Techniques such as software interface control meetings and the generation of baseline software interface control documents should be a necessity on all programs.

14. Recognize interoperability considerations. Most weapon systems must function in a multi-system environment. Adequate attention and resources should be applied to develop inter-system interface standards early in the specification process, to establish configuration management methods to control them, and to develop realistic test methods for validating them. Interoperability problems in many tactical information systems involve costly software changes if corrected late in the process.

15. Maintain user involvement as the design progresses. Since the user (including both operations and maintenance) will be required to 'own' the system after acceptance, minimize the number of surprises or operational objections to the system by maintaining a constructive but well-controlled interface with representatives of user groups.

16. Establish a separate resource for the monitoring and validation of software development activities. An identifiable resource should be assigned to monitor and validate the activities of the software developer when significant amounts of software are involved. Use of in-house laboratory software personnel or a separate software validation contractor to supplement the project office should be considered.

17. Software integration and test and evaluation facility. A software integration and test and evaluation facility should be planned for and available for software integration testing early in the software development cycle. Special software and hardware required to develop this facility should be included in the initial contract arrangements. Where feasible, facilities and personnel should be shared between weapon systems.

3.1.2 Supporting Studies and Analyses

Recommended Action: Require specific software design tradeoff studies and analyses as part of the performance specification process.

Certain supporting tradeoff studies and analyses should be conducted during the early definition of software and system requirements and during the early software design formulation activities. Three studies and analyses were mentioned in Section 3.1.1 and are further discussed here:

1. An analysis of the proposed software design approach versus the use of hardware or other design approaches.

2. An analysis of new software developments and facilities versus the use of existing resources.

3. An analysis of the software development risks involved.

These three analyses should be conducted, as a minimum, when significant levels and complexity of computer hardware, software, and costly support facilities are involved. Other tradeoff studies and analyses may be required because of special requirements or risks associated with a specific weapon system development.

3.1.2.1 An Analysis of the Proposed Software Design Approach Versus the Use of Hardware or Other Design Approaches

This analysis is required to insure that the chosen software and hardware design represents the most cost-effective approach for satisfying the mission requirements when all factors are considered. The major factors should include:

- Consideration of computer hardware and software versus hardwired logic, firmware, and/or manual (procedural) alternatives.

- Consideration of both operational and support software areas.

- Comparison of associated life cycle costs.

- Consideration of the benefits of a software approach where future mission changes can be expected.

- Consideration of User preferences.

- Evaluation of associated development and technology risks.

- Consideration of performance and reliability/maintainability tradeoffs.

A separate report presenting the results of this analysis should be prepared and should be available to support the decision for entering full-scale software development.

3.1.2.2 An Analysis of New Software Developments and Facilities Versus the Use of Existing Resources.

This analysis is required to ensure the efficient utilization of existing DoD computer hardware, software, and facilities before initiating new developments and establishing new support facilities. The major factors should include:

- Availability of off-the-shelf computer hardware designs which satisfy computer performance and capacity, physical, environmental, and reliability/maintainability requirements.

- Availability of software packages -- including operational (application) software, operating systems, development and maintenance support software (e.g., compilers, assemblers, utility routines), and operational support software (e.g., automatic test equipment and diagnostic software, training/simulation software) -- which satisfy applicable portions of mission requirements and which are transportable (i.e., have demonstrated performance, are adequately documented, and most important, that trained personnel are available to assist in the transition to the new project).

- Availability of design support, integration, test and evaluation (validation), and maintenance support facilities, used for similar weapon systems which have an unused capacity.

- Consideration of sole source and procurement implications.

It should be noted that 'portability' of computer hardware and software between projects to date has been largely limited to standardized families such as the UNIVAC AN/UYK-7 and AN/UYK-20 and the IBM 4 Pi series, to support software areas (such as compilers, assemblers, and utility systems), and in some instances to operating systems. However, the cost and schedule benefits to be achieved by utilizing existing resources can be significant and DoD should emphasize efficient utilization wherever practical.

A separate report presenting the results of this analysis should be prepared and should be available to support the decision for entering full-scale software development.

3.1.2.3 Analysis of the Software Development Risks

This analysis is required to ensure that adequate risk management methods (strategies) are applied for software where significant development risks are involved. The major factors should include:

- Mission requirement uncertainties which might impact the software.

- Likelihood of significant user changes and additions during the design formulation phase.

- Development risks associated with the level and complexity of the software and the system architecture.

- Computer hardware and interface dependencies.

- Experience of major participants.

- Location and availability of adequate resources and facilities.

The risks should be assessed in terms of cost, schedule, and mission performance impacts. A separate report presenting the results of this analysis should be prepared and should be available to support the decision for entering full-scale software development.

3.1.3 Requirements Development and Validation Methods

Recommended Action: Establish formal software quality assurance practices which require the use of proven software design, development, and validation methods.

Formal DoD-wide software quality assurance (QA) practices should be consolidated and should be consistently applied to the contractor's activities during the software development and validation process.[1,2,3] The specific objectives of the software QA program should:

1. Ensure that the design of the delivered operational software elements (software packages and related documentation) conforms to good design practices and to the design objectives agreed upon at the time of contract.

2. Ensure that the performance of the operational software elements when integrated with the hardware and external interfaces conforms to: (1) the specific software performance standards (quantitative values) agreed upon at the time of contract; (2) the operational (functional) requirements stated in the software development specifications; and (3) the real requirements of the user and intent of the overall weapon systems mission requirements.[4]

[1] Portions of the material in this section have been extracted from MITRE Technical Report MTR-6906, Software Quality Assurance and Production Control Practices in the Acquisition of Large Systems. It is included here to provide management with a flavor for the nature and the level of detail of the practices and resources required to establish a valid software QA program. The reader should refer to this referenced report for a more thorough discussion of software QA and production control practices.

[2] The developer is referred to as the contractor in this discussion. However, the principles also apply where the developer is an in-house resource.

[3] Note: The discussion on software quality assurance addresses more than the methods and techniques needed to develop and validate software requirements. It includes all topics normally associated with software QA; i.e., all the government's activities required to reach a valid design and to ensure that quality software products are delivered on time and at agreed-upon cost. The discussion is presented here because of the need for DoD to address QA as a coordinated action area.

[4] This objective is generally synonomous with the accepted definition of Verification and Validation (V&V) in DoD.

3. Ensure that the system will function (interoperate) effectively in a multi-weapon system environment.

4. Ensure that the operational software elements are developed, merged with the other system elements, and are accepted within the cost and schedule objectives as stated in the contract and in the system management plans.

5. Provide resources and methods for timely resolution of contractor's design questions, proposed changes, and development alternatives.

6. Identify to management as early as possible in the process software related problems and resource deficiencies that might impact the above objectives.

7. Maintain strict controls over the system functional requirements and design (freeze as early as practical - ideally as close to the critical design review as possible) and minimize the number of changes during the course of the contract that might impact the above objectives.

8. Ensure that adequate support software, system resources, and related documentation are included with the delivery of the system to satisfy operations and maintenance support functions.

To meet these objectives, specific QA activities and resources should be applied during the contract. The activities should be in addition to those provided by the contractor's own software QA program and should be organized and put into practice in such a way as to assist and supplement the contractor and not hinder or impose an excessive workload on him. An open and constructive interface between the contractor and the government's representatives is a prerequisite to the successful implementation of a QA program.

Several software QA practices which should be considered in the development of the DoD software QA guidelines include the following:

1. Those required to ensure a quality design, such as:

   - Establish early software and documentation design standards.

- Establish a central software design file (e.g., notebook or library) which centralizes important software design and status information.

- Allow access to the design file and other design documentation by government QA representatives.

- Establish and allow access to results of software modeling and sizing activities.

- Provide for early and periodic review of contractor's software design approach.

2. Those practices required to control software requirements and ensure acceptable performance, such as:

   - Review the software design to ensure that system requirements and mission intent are being met.

   - Impose a design freeze after the design reaches an acceptable risk level. As a general practice, add new changes or additions as future packages.

   - Periodically participate in modeling and testing activities and review results.

   - Establish formal software configuration control procedures early which become increasingly more stringent as the acquisition process proceeds.

3. Those practices required to ensure on-cost, on-schedule delivery, such as:

   - Require periodic project status reviews.

   - Establish software progress milestones chosen to show tangible evidence of progress.

   - Require an up-to-date development plan.

   - Monitor early testing to validate the contractor's (often overly optimistic` progress estimates.

   - Provide fast response to the contractor's action requests.

   - Require periodic and open QA review meetings.

   - Require periodic and frank QA reports to management.

4. Those practices required to define and conduct in-plant and on-site acceptance of software products, such as:

- Require QA review/approval of test documentation.
- Participate in in-plant testing and require formal in-plant acceptance of applicable software elements.
- Participate in on-site validation testing with direct user participation.

Not all of the above would likely be imposed on a single project. Rather, the level of QA activities and resources should reflect the uncertainties and risks involved.

The selection of a separate software validation contractor to basically perform the above QA tasks was being followed by several of the project offices visited. This approach, as well as the use of in-house laboratories to supplement the project office QA personnel, should be considered in the development of formal QA guidelines.

3.1.4 Management Controls Over the Performance Specification Process

Recommended Action: For 'major' systems, require specific software supporting documentation and analyses as part of the OSD level DCP/DSARC review process. Require similar supporting documentation and analyses at Service levels for 'non-major' systems or systems past the equivalent DSARC III decision point but involved in a major update cycle.

The preparation of comprehensive DoD guidelines covering the software performance specification, development, and validation process (such as those discussed in the preceding sections) does not insure that these guidelines will be imposed on the future activities, nor necessarily followed by all project personnel. Specific controls (checks) must be established at OSD and Service levels which require that these practices be followed. Immediate action is needed to ensure that the efforts leading to the preparation of the Decision Coordinating Paper (DCP)[1] and its

[1] As discussed in DoDI 5000.2, "The Decision Coordinating Paper and the DSARC".

subsequent updates take into account the software performance specification, development, and validation factors which need to be considered, and to ensure that necessary support documentation on these aspects of software is prepared and available for timely review. Similar action should be taken to ensure that similar factors are also considered in 'non-major' systems and/or major updates to systems past the equivalent DSARC III decision point.

Current DoD Directives and Instructions exist (e.g., DoDD 5000.1 and 5000.26, and DoDI 5000.2) which require review information on a system basis. Many of the information requirements of these Directives/Instructions are applicable to weapon systems software. However, because of the lack of weapon systems software definitions, software acquisition process structures, software work breakdowns, etc., the software is not generally subjected to reviews to the same degree as other system components. To initiate software reviews, immediate action should be taken to require the analyses and the submission of the following types of DCP support documentation.

1. A report presenting the results of an analysis of the proposed software design approach versus the use of hardware or other design approaches (this analysis is described further in section 3.1.2.1)

2. A report presenting the results of an analysis of new software developments and facilities versus the use of existing resources (this analysis is described further in section 3.1.2.2)

3. A report presenting the results of an analysis of software development risks (this analysis is described further in section 3.1.2.3)

4. A software acquisition plan which addresses in one document all of the important software performance specification, development, and validation factors previously described in section 3.1.1, as well as the software acquisition planning factors described in section 3.2.1.

All of the above items should be prepared and/or updated in support of each DCP/DSARC decision point (or equivalent 'non-major' decision point), where possible. Where data is not available to prepare all areas of these reports (such as at the initial DSARC I decision point), separate justification as to why it is not available should be presented.

3.2 Software Acquisition Planning

The corrective actions in this area are concerned with the recognition and consistent application of sound management practices to the process of acquiring software end products. The practices are intended to provide management awareness and visibility over the software acquisition process and to provide controls (checks) that ensure the consideration of all important factors.

3.2.1 Checklist of Important Software Acquisition Planning Factors

Recommended Action: Identify the important weapon systems software acquisition and life cycle planning factors and establish specific DoD guidelines to ensure that these factors are adequately considered in the software acquisition planning process.

The lack of management emphasis, awareness and visibility over software activities has been identified as a major contributing factor to problems in weapon systems. In the past, there has been a tendency of management to emphasize hardware portions of systems first, leaving software until last. This approach is often inconsistent with the critical role played by software subsystems. While improvements to specific management practices are being pursued actively by the Services, and models of comprehensive planning are evident in several new systems, it was noted that they are still not being applied consistently across all software-based weapon systems.

This section lists several important factors which should be formalized during Phase II of the study. It is not intended to be a complete list, but rather to indicate the nature of the required DoD guidelines. Several items are further expanded in later sections.

1. Require that a documented software acquisition plan exist early in the process and that it be periodically updated at key decision points. In weapon systems where significant levels of software are involved and/or where software is critical to the overall mission success, a separate, documented management plan should be required specifically for the software components. Its content should be as comprehensive as possible and as a minimum, should include all items discussed in this section.

2. Identify total software life cycle requirements and establish an approach for their orderly development. The early planning activities should identify all major software end items and resource requirements over the total expected life cycle of the system, and should provide a planned approach for their orderly development and availability at the required times in the program. The software end items should include, as a minimum, operational software, support software (e.g., software development tools, test and validation software, and operations and maintenance support software), automatic test equipment and diagnostic software, and training/simulation software. The resource requirements should include software-related facility requirements such as system integration and test and validation facilities. The planning should specifically address software operational and maintenance requirements (i.e., software and related support facilities required after system transitions to user and maintenance commands) and should describe how these requirements will be satisfied during the development phases.

3. Require an analysis of software development risks. An analysis should be conducted to assess the risks involved with the development of software for the weapon system. The operational software architecture and new or unique software that needs to be developed should be analyzed along with difficulties that may be encountered due to mission and requirements uncertainties, software size, involvement of many organizations, contractor risks, geographically separated facilities, etc. The risk should be assessed in terms of cost, schedule, mission performance, reliability, and maintainability.

4. Establish an overall software acquisition and procurement strategy. When a significant amount of software development is (or is expected to be) involved in a weapon system, a specific software acquisition and procurement strategy should be developed at the time of program initiation. This strategy should consider the software development risks involved, methods of providing contractor incentives, dependencies between software and other major subsystems, and overall schedules and methods for expediting them.

5. Utilize software prototyping and/or parallel developments where significant risks or requirements uncertainties exist. Extensive use should be made of software prototyping and/or parallel developments when software development risks exist or user requirements are uncertain. In some cases,

use of existing resources in existing systems should be used to demonstrate user requirements or to simulate performance before entering into costly long-term software developments.

6. Establish specific development phases for software. There are significant differences between the phases for software development and those of hardware and the weapon platform. These differences should be recognized and specific phases identified for software.

7. Establish specific decision points for software. For some weapon systems, the DSARC decision points will align to the weapon platform milestones rather than to software (e.g., in an aircraft or missile). In such cases, there may be a tendency to deemphasize software other than that needed to 'fly' the platform during the initial validation phase (fly off). In these cases, separate intermediate software reviews and decision points are required.

8. Establish specific software progress milestones. The tendency may be to concentrate on the resolution of hardware-related problems during the initial development period and not to emphasize software until it is merged with the hardware. Specific software milestones should be established which reflect software schedules as well as overall system schedules and be closely monitored. Emphasis should be on choosing intermediate milestones that provide tangible evidence of progress.

9. Require reporting of specific software management information and thresholds. Periodic reporting of specific software cost, performance, and schedule information should be required. This information should be in a format which allows management to measure progress against established management cost, performance and schedule goals. Thresholds should also be included to alert management in the event of trends that may lead to software cost overruns, performance degradation, or schedule impacts.

10. Identify roles and responsibilities of all organizations as early as possible. The development responsibilities and the source of all resources and facilities should be agreed upon early in the acquisition process.

11. Use separate validation resources. The identification and use of a separate validation resource (e.g., contractor or in-house laboratory) should be considered to improve management's visibility of software.

3.2.2 Definition of Acquisition Phases and Milestones Specifically for Software

Recommended Action: Define specific acquisition phases and milestones for weapon systems software which reflect the true nature of software development in the overall system acquisition process. Define related guidelines for use by project planning personnel.

Three major concerns involving the definition of specific phases and milestones for software were noted in referenced DoD studies and repeated in several of MITRE's weapon system interviews. They include:

1. A lack of management emphasis on software during the initial weapon system phases (i.e., a tendency to provide resources to start software late).

2. A tendency to align software to the phases and schedule constraints of the weapon platform (e.g., the missile or aircraft) and to the DSARC decision points rather then to realistic software phasing requirements.

3. A lack of proven methods and milestones which can be used by project personnel to gain visibility into the contractor's software activities and to show tangible evidence of progress.

The development of DoD guidelines in this area should include the following important factors:

1. The need for separate and intermediate phases, milestones, and decision points for software from that defined in the 5000 series publications and which are applicable to the wide diversity of weapon system software types.

2. The need for early emphasis (i.e., management and fiscal resources) for software.

3. A recognition in allocation of resources and phasing of the overall system that the software process is often compacted. This is because of the need for software concept validation and design activities to follow the overall system concept validation and design process, and the need for the final software development phase to precede that of the system production phase.

4. A recognition of the long lead times often associated with the development of computer hardware and software elements.

5. The need for a period of software integration with the hardware and the need for a redesign iteration in which software changes can be expected.

6. A need for the project participants (including project office, user groups, and new contractors) to become educated on the system and mission requirements. (For example, don't expect a new contractor to be an expert artilleryman, nor a user group to understand the subtleties of a contractor's design approach.)

3.2.3 Definition of Acquisition Strategies Specifically for Software

Recommended Action: Define specific weapon systems software acquisition and procurement strategies (such as software prototyping and parallel development) which maintain contractor incentives and limit software development risks. Define related guidelines and criteria for use by project planning personnel.

Many of the problems discussed during the study originate in the need to develop specific software acquisition and procurement strategies which limit software development risks and which maintain industry incentives further into a project. The development of guidelines and criteria in this area should consider the following important factors:

1. A need to develop strategies which address the following types of software development oriented issues:

- In what order will software be developed?

- In what increments will the software be completed and delivered for test and use?

- What provisions are made in schedules and in allocation of resources for iterating the design of each major component, and how does this tie in with total system development time planning?

- What are the major milestones, and what provisions are made in the management process to alert management as problem trends start to occur?

- What concurrent developments are planned (e.g., software with computer hardware; compilers with programs to be compiled)?

2. A need to develop strategies which address the following types of acquisition planning and management issues:

- What provisions will be made for reducing identified software development risks?

- Will each software subsystem be developed in-house, by one or more contractors, or by a combination?

- How can industry competition be maintained throughout development?

- Will there be a software prime contractor or several associate contractors, and how will software responsibility be divided?

- What will the technical basis for the contract(s) be? Will a detailed specification be used or will the developer be given design freedom?

- Will there be a testing and verification capability separate from the software development organization(s), and will this be government or contractor and under whose control?

- How much off-the-shelf software will be used, and is it really available?

3. A need to incorporate results (lessons learned) from previous program experiences, such as:

- Avoid concurrent development of computer hardware and software unless there is a plan for several major iterations of both the hardware and software design.

- Avoid development of support software (e.g., compilers and operating systems) concurrently with applications software.

- Allow sufficient time and resources for design, design iteration, and evaluation early in a development. For example, in a contract, don't expect a completed software design for any substantial program in 90 days.

- Plan for early demonstration of a small increment for test and use, followed by incremental demonstration of additional capabilities.

- Develop milestones appropriate to the project at hand and consistent with overall system development time phasing.

- Avoid dividing responsibility for concurrent, related software development among several different organizations.

- Don't begin a full-scale software development until the buyer, his technical advisor, the developer and the user are satisfied with the specification.

Four specific acquisition and procurement strategies were discussed during Phase I of the study, both within MITRE and with different DoD personnel. They reflect types of strategies that can be used to partially overcome many of the software requirements, cost, and schedule growth problems that impact many DoD systems. While none are cure-alls, they should be given serious consideration in developing software acquisition guidelines during Phase II of the study. Each is very briefly discussed in the following paragraphs.

1. Provide for early simulation of user requirements during the contract definition phase. Many of the software requirements changes introduced by the user were found to be late in the program after the initial system was built and available for user test and evaluation. Many of these changes involved requirements that could have been simulated and demonstrated to the user early in the project through the use of existing general purpose peripherals, computers, and graphic displays. In some instances, an earlier generation of a weapon family could have been used to conduct controlled experiments to arrive at requirements with a higher confidence level.

2. A formal software prototype phase should be required where risks and uncertainties are involved. The generally accepted software approach is to attempt to develop software once which is unlike hardware where formal brassboard, prototype and (pre) production phases are generally followed. The built-right-the-first-time philosophy is often not successful in practice. Software prototypes are used here to define computer programs which perform some or all of the functions intended for the system, designed and built with minimum support and documentation to save time and costs, and intended to answer specific questions. For the most uncertain case, three successive kinds of prototypes can be distinguished.

- A 'functional' prototype or brassboard intended to demonstrate that the software performs the functions expected by the users and developers. This program may be built on a computer other than the one intended for system use.

- A 'performance' prototype or engineering prototype intended to demonstrate that the software uses the expected amount of storage and delivers the required throughput with the required response times. This prototype is built to run on the system computer.

- A 'production' prototype intended to demonstrate that the software is designed and operates in such a way that it can be adapted to different sites and that software maintenance can be performed.

For limited software developments or software developments based on previous systems and for which specifications can be written directly with high confidence, one or more of these prototypes may not be necessary. In many cases, major portions of the prototype software can be used in the final version.

3. Establish a parallel software development during the initial program phases. Development of two versions of a system in parallel to reduce the risk that one might not be feasible, satisfy mission reouirements, or be completed on time is a well-known risk reduction and incentive technique. The use of similar techniques for software should be considered. The intent of parallel software development should be primarily to assure an alternative source of software and as a contingency against failure of a software development to achieve its objectives.

4. Establish a formal four-step software acquisition process which assumes the software to be evolutionary (will change during the program) and addresses the issues accordingly. There are two obvious extremes in controlling software requirements growth and the resulting cost and schedule impacts: 1) to mandate strict controls over the user after the requirements (contract) definition phase; and 2) to assume change will occur and provide for a modular design which allows for change.

The desired solution should include components of both extremes. The four-step process discussed here is aligned to respond to the user's requirements as well as to the developer's real problems which occur when uncontrolled changes are allowed to be incorporated freely. The highlights of the four phases include the following:

Phase I, Contract Definition. The user and project office takes the lead in developing the Type I specifications. The contractor(s) is on board only in a supportive role. The products from Phase I are software specifications (a frozen set of requirements) which the user agrees upon.

Phase II, Prototype Development. The contractor develops and delivers a prototype software design based upon the Phase I frozen design. No changes are allowed (in principle), but the user and project office understand that changes will be incorporated in Phase III before a field version of the software is produced for operational use. The product from Phase II is a sound test vehicle for Phase III use, developed with minimum interaction from the government.

Phase III, Test and Evaluation. The user, project office, and developer participate in a test and evaluation phase to arrive at the final Type II specifications.

Phase IV, Software Production. A final field version of the software is produced based upon a frozen Type II specification. Subsequent changes are incorporated as field versions.

While many difficulties can occur, such as a complete change in the overall mission requirements during Phase II, the approach has merit and should be considered further.

3.2.4 Common DoD Definitions for Software Terminology

Recommended Action: Establish common definitions for software terminology for use throughout the DoD and by DoD contractors.

There is a need to establish a common set of definitions for weapon systems software terms for use within the DoD and by Defense contractors. Common definitions are required to improve understanding and communications when addressing software acquisition, maintenance and management processes. Common terms are also needed in other areas, such as development and maintenance of a computer programs catalogue, and improving the transfer (portability) of software between weapon systems.

The term 'Weapon System Software' by itself is not well defined and it assumes different meanings, depending upon the subject at hand or the individual using the term. Definitions for hardware acquisition have evolved through the long history of hardware development and procurement. Attempts to define software within existing procurement terms and categories often causes confusion and subjects it to inappropriate regulations. One example of the problem was found during the weapon systems interviews where representative cost information could not be collected across weapon systems, partly because of the lack of common definitions for the components of software.

Another example is the fact that software categories are not well defined. The categories frequently used are: operational software, automatic test equipment software, training/simulator software, development/production support software, testing support software, and maintenance support software. Clear definitions of these and other terms will be useful for contract definition, identification of deliverables, and software acquisition planning and management.

The current DoD Steering Committees activities concerned with establishing common definitions for basic software terminology, such as 'computer data', 'computer program', and 'computer system', should be encouraged. Further, these activities should be expanded to provide accepted DoD-wide definitions and breakdowns for broader software related subjects such as:

- Categories of computer based systems (e.g., weapon system, C^3, intelligence).

- Categories of software required to develop systems (e.g., design and analysis tools, development tools, test and validation tools).

- Categories of software integral with a system (embedded software) and that required to support their operation (e.g., operational flight programs, automatic test equipment and diagnostic software, crew training and weapon simulation software).

3.2.5 Exchanging/Sharing of Software Resources Across Programs and Services

Recommended Action: Establish methods for identifying DoD resources applicable for use across systems and Services, for example, through the preparation and maintenance of a catalogue (inventory) of weapon systems computer hardware, software, and facility resources. Define related guidelines for use by Service level and project personnel.

The efforts initiated through the DoD Software Steering Committee to develop a catalogue of computer hardware and software used in weapon systems should continue. The OSD, the three Services, and the Marine Corps should continue to participate in this activity. The catalogue, however, should be expanded to include a description of major facility resources that could be considered for shared usage. The catalogue should contain sufficient information so that an organization can recognize whether it would be worthwhile to obtain additional information about computer hardware, software, and facilities from the development or user agency to satisfy a requirement for a new or existing system. In developing the catalogue, recognition should be given to the various software categories for weapon systems. The exchange and sharing of hardware, software, and facilities can achieve savings in both time and effort by software personnel and in the costs of hardware and facilities. Direction and guidance should be issued for maintaining the catalogues and requiring the Services to review the catalogue information for possible use of existing hardware, software, and facilities for new acquisitions.

In addition to using the catalogue to determine the availability of software resources, it can be used as an aid in developing standards. For example, reviews could be made of the catalogue, and records maintained on the frequency of use of existing hardware and software, to form a basis for determining the appropriateness of standardization to minimize future unnecessary duplication. The catalogue could also be used to identify areas of possible software development which could be of most benefit within a Service or throughout the DoD community. For example, it may be that software development could be initiated to provide a common set of support software for a family of computers that is frequently used across DoD.

3.2.6 Software Management Information

Recommended Action: Initiate OSD action to require the collection and dissemination of selected weapon systems software management information including software-related cost, performance, and schedule information.

In order to improve management's visibility and awareness of software in weapon systems and to provide a basis for future management policies, OSD should initiate action to define a minimum set of software management information measures and develop procedures and practices for their use.

Guidelines for the Development of Software Management Information Measures

The definition of DoD software management information measures and procedures and practices for their use should follow certain constraints or they may prove counterproductive. Imposing new data collection and reporting requirements will impose new workloads and create additional review levels, and should only be approved where clearcut benefits can be realized. To avoid imposing excessive reporting requirements, the following ground rules (or constraints) should be followed:

1. The DoD philosophy should continue to be one of decentralized project management with appropriate upper-level controls. That is, the program managers should be allowed to use their discretion in the day-by-day management of individual programs as long as the project goals are not compromised.

2. Upper level management information should be derived as a subset and/or interpretive set of lower level requirements; i.e., required OSD information should be a subset of that prepared for service level reviews.

3. All data reporting requirements must be on a need-to-know basis; i.e., each reported data point should be tied to a definite need or benefit.

4. The number of data points should be kept to as few as possible with the reporting periods kept as long as possible and still be useful (i.e., remain sensitive to critical problem trends).

5. The measures should be common across the acquisition phases and across systems where practical.

6. New procedures for software information flow and review should use established methods and instruments where practical (e.g., supplement the supporting data provided under DCP/DSARC policies rather than develop new points).

7. Thresholds should be established for software costs, technical performance, and schedules which require immediate alerts to higher levels when trends which may lead to the thresholds being exceeded are identified. This approach limits the reporting instruments to major review breakpoints and large reporting periods, yet alerts higher management immediately to software problem trends.

Proposed List of Initial Software Management Information Measures

Certain cost, technical, and schedule information measures can be defined now on the basis of past experience on large software projects. Others, largely in the areas of performance and quality (e.g., reliability, maintainability, portability, productivity) will require further study and use in pilot programs before they can be used universally across the DoD.

A proposed minimal list of information measures that can be defined now and which should be considered for early DoD use is described in Table 3-1. They should be considered only as an interim solution, to be replaced by more sophisticated measures when they are adequately defined and validated. Consideration should be given to an early pilot application of this set of measures to one (or several) system(s) to better determine their effectiveness and utility versus added project office costs and workloads imposed.

TABLE 3-1
PROPOSED LIST OF INITIAL SOFTWARE MANAGEMENT INFORMATION MEASURES[1]
(RECORDS KEPT AND RETRIEVABLE AT EACH LEVEL FOR EACH SYSTEM0

MEASURES	OSD	SERVICE	SPO (AND SYSTEM MANAGER AND USER AFTER TRANSITION)
COST	● Yearly Software Costs Broken Down by: - System • Software Contracts Total • In-House Software Facilities Personnel Total • Actual Versus Estimated ● Five-Year Software Cost Projection for Above. ● Thresholds for Above (Alerted if Expenditure Rate Exceeded).	● Yearly Software Costs Broken Down by: - System/Element • Software Contracts Totals • In-House Software Facilities and Personnel Totals • Actual Versus Estimated ● Five-Year Software Cost Projection for Above. ● Thresholds for Above (Alerted if Expenditure Rate Exceeded).	● Quarterly Software Costs Broken Down by: - System/Element/Subelement • Software Contracts Totals • In-House Software Facilities and Personnel Total • Actual Versus Estimated ● Five-Year Software Cost Projection for Above.
TECHNICAL	● Yearly Review of Growth Factors/Budgets (past, projected). ● Lost Time Threshold (Alerted if Lost Time Due to Software Exceeds a Set Threshold Agreed to Between OSD and Service).	● Six Month Review of Growth Factors/Budgets. ● Threshold for Above (Alerted if Projected to be Exceeded by 10%). ● Lost Time Threshold (Alerted if Lost Time Due to Software Exceeds a Set Threshold Agreed to Between Service and User).	● Growth Factors Broken Down by: - System/Element/Subelement • Storage/Capacity Budgets • Response Time, Processing Time Budgets ● User Lost Time Due to Software Broken Down by: - System/Element
	● Completion of Major Software Mile-Stones by Element. ● Thresholds for Above (Alerted if Projected Slippage Exceeds a Set Threshold Agreed to Between OSD and Service).	● Three-to-Six Month Schedule Milestones Broken Down by: - System/Element ● Thresholds for Above (Alerted if Projected Slippage Exceeds a Set Threshold Agreed to Between Service and Project Levels).	● Detailed Schedule Milestones Broken Down by: - System/Element/Subelement

[1] Examples of Software Breakdowns Used in This Table:

System - Refers to All Software in System as a Single Line Item.

Element - Refers to Major Categories of Software (e.g., Operational Software, Automatic Test Equipment Software, Support Software, etc.).

Subelement - Refers to Major Packages of Software Within an Element (e.g., Different Software for Different ATE).

Management Weapon Systems Software Cost Model

In order to understand total software costs across all DoD weapon systems, a methodology should be developed for identifying the components of software costs and for collecting and analyzing the necessary cost data. This type of accumulative cost information is required in addition to the previously discussed types of information in order for management to understand where major portions of total costs are expended and where management emphasis and action are required.

Future efforts to determine the total cost of software in weapon systems should include (start with) the development of a management cost model and agreement on its content. The model should define the software cost components and should identify which defense systems, personnel, and facilities are applicable. One such a model[1] is agreed upon, more meaningful data can be collected and total cost estimates derived.

3.2.7 Review of Major DoD Publications Governing Software Acquisition

Recommended Action: Review major DoD publications (i.e., directives, instructions, regulations and MIL standards) used in the acquisition of software in weapon systems. Initiate interim changes to correct for software omissions, deficiencies, and conflicts until formal long-term solutions are implemented.

Review of current DoD publications was not possible within the scope of this study. However, sufficient concern was expressed during the interviews and from a limited review of selected publications to indicate that an in-depth review is needed. The reviews should be conducted from at least two aspects: (1) to extend good hardware policies, procedures and practices to software where they are equally applicable; and (2) to expand publications to cover weapon systems software acquisition management more thoroughly where omissions and conflicts exist. In

[1] In Appendix F of this report represents a possible management weapon system software cost model. It is intended to serve as a possible starting point for future DoD efforts in this area.

conducting the review, the need for a separate DoD-wide publication for software acquisition management should not be overlooked. Examples of areas where changes are needed include:

1. MIL-STD-490, Specification Practices

a. Maintainability considerations are required for hardware, but no mention is made for software.

b. Design approaches and the use of design and support software are not addressed.

c. Recognition should be made that different categories of software exist, (e.g., operational, support, testing, automatic testing equipment, and training simulators).

2. MIL-STD-881, Work Breakdown Structures (WBS) for Defense Material Items

a. Expansion of the software work breakdown structure is needed to facilitate better cost estimating and control, proposal evaluations, contract negotiations, and software acquisition scheduling and management. Although some information is included for computer programs under the Electronics Category, the definitions should be expanded, and other types of weapon systems software should be covered.

3. DoD 5000 Series Directives and Instructions

a. Ensure that wording throughout the DoD Directives and Instructions dealing with weapon systems acquisitions makes it clear that certain practices apply to software as well as to hardware. For example, in DoD 5000.1, Acquisition of Major Defense Systems, paragraph III. C.1 should be updated to include software where it discusses meeting operational needs through use of existing military or commercial hardware. Another example is in paragraph III. C.5 of DoDD 5000.1 where it discusses the use of models, mock-ups and system hardware to increase confidence levels; the wording should also refer to software.

4. General

The phasing for development and reviews, defined in the various directives and standards, are generally for hardware. For example, the preliminary design reviews required by military standards are generally too early in the system process for software.

3.3 Software Technology

Recommended Action: Ensure that research, studies, and pilot programs are initiated or continued in areas where current technology and management practices are inadequate in meeting the requirements for efficient development of reliable software and for effective management control of the development process. Eleven areas are discussed which should be given high priority in DoD allocations of software R&D funds.

This area consists of R&D, study and pilot programs to provide for a continued and coordinated software technology program to further improve and develop the management methods, practices, and techniques for software development. The DoD should ensure that research is initiated or continued in areas where current technology is inadequate in meeting the requirements for efficient development of reliable software and for effective management control of the development process. Where more information is needed, studies should be conducted. When solutions have been proposed, opportunities should be provided to apply and evaluate these solutions in pilot programs which present real military problems. After their feasibility has been demonstrated, the results can be used to alter regulations, standards, and practices for software acquisition. Programs which should be given high priority include the following areas. Some of these areas are ongoing and should be continued. Others will require the initiation of new projects.

3.3.1 Develop Quantative Engineering Measures for Software

This activity involves the development of quantitative measures of the status of software and its reliability which can be used to monitor and predict progress toward schedule and performance goals. There is a lack of visibility in the software development process and a resultant uncertainty in meeting schedules and satisfying quality criteria because of a lack of quantitative measures of software status and quality. To develop such measures requires accurate data on time spent in various phases of software development as a function of software attributes such as size and complexity, language, and tools used. Error rates must also be accumulated for use in verifying models for predicting software reliability. Automated aids for collecting and analyzing the data will be necessary for the development and verification of measures. A research program to determine measures would include collection of the requisite data, synthesis of measures such as errors-per-line-of-code and lines-per-day-per-programmer, and tests of these measures on real software development.

3.3.2 Define Characteristics/Methods for Improving 'Portability' of Software

This activity involves the continued research by the Services to improve methods of developing transportable software, capable of being executed in more than one operating environment. This action will support the wider use of software inventories and will influence decisions on the impact of standardization. If software developed for one operating environment (i.e., one computer type, operating system and set of interfacing software) could be used in similar but not identical environments, costs for developing different versions of similar software packages (e.g., compilers for the same language) could be reduced. Techniques for creating transferable software are being studied and developed by such groups as FIPS Task Group 13 and the Navy ADP Selection Office (ADPESO). These methods should be adapted to weapons systems operational and support software. Current procedures for languages such as FORTRAN should be expanded to include manual and automated techniques needed to ensure transferability. Problems with current automated aids should be identified and eliminated. Languages such as TACPOL (Army) and OPAL (DoD test equipment) should be studied now to eliminate features which may cause future transferability problems. Operating system command languages should be analyzed to determine features which inhibit transferability.

3.3.3 Investigate Automatic Programming Methods with Emphasis on Improving Correctiveness of Software

This activity includes the identification of new methods to develop more reliable software. Research should be continued into new approaches to the entire software development process, such as automatic programming, with specific emphasis on automated aids to provide for the correctness of software. Many separate research activities are being sponsored by DoD in Automatic Programming including both automated aids to programming and the fundamental technology of 'knowledge-based' systems. These techniques are expected to increase the efficiency of the entire programming process, from statements of software requirements through testing, by aiding or performing the analysis of requirements, the automatic generation of code, and automated testing. This long-term effort requires fundamental research with centralized coordination and monitoring to determine when advances to pilot stages are feasible. Emphasis should be placed on the automated side to testing software because the highest cost (almost half the development cost) is estimated to occur during the validation process.

3.3.4 Develop Methods for Predicting Cost of Software

This study activity should be supported by the Services and involves the development of models for use in predicting the cost of software at various stages in system acquisition. Software cost estimates and predictions will be more accurate and more credible if they are based on a substantial collection of accurate cost data. The distribution of present and anticipated software costs should be determined as a function of activity, type of application, and life-cycle phase. Data to be collected must be defined and a data collection point designated. A necessary input to this activity is actual cost data. Data should be collected on DoD in-house costs as well as contractor costs. Sufficient descriptive information should be collected to allow categorization by software type, strategy used for development, relationship to computer hardware, and similar parameters. Case studies of acquisition programs should be funded to isolate useful and collectable cost data. Cost data elements required should be proposed and data collected on a representative sample of programs. Based on this experimental effort, cost data collection procedures and cost estimating methods should be developed to build a continuing cost data base. The data base would be used in cost estimating, measuring improvements, and challenging over- and under-estimates of predicted software costs.

3.3.5 Investigate Benefits/Methods for Effective Standardization of Programming Languages and Support Software

This study activity should be continued by the Services to determine the areas and methods for effective standardization of programming languages and support software (e.g., compilers, assemblers, etc.) so that appropriate standards can be adopted. Standardization, properly effected, reduces the number of different items to be produced, the size of necessary inventories, and extends the scope of application and the life of standardized items. To effect software standardization, a study must first identify those characteristics of programming languages and support software which are critical to standardization decisions. The study will then determine which aspects of software make it different. Data gathering efforts, similar to the collection of functional and language requirements being effected by the Air Force HOL Standardization Program should be carried out DoD-wide to identify areas of software acquisition and development in which standardization can be applied effectively. (The preparation of a DoD catalogue of computer hardware and software discussed in Section 3.2.5 would provide additional data for the standardization efforts discussed here.) Previous standardization efforts which have failed should be studied as well as those which have succeeded (such as the CORAL standardization program in the United Kingdom) to extract lessons for future standardization. The managerial and executive functions and organization required for successful implementation of standardization plans should be determined. Finally, the development of an evaluation methodology, which can predict the impact of a contemplated standardization action or can assess the impact of a completed action, would provide valuable assistance in decision making and in refining standardization criteria.

3.3.6 Conduct Software Methodology Pilot Programs

This activity would be accomplished by the Services by conducting pilot programs which consolidate and apply advanced techniques and tools to the development of software for real military applications. Results should be evaluated and disseminated across Services and to industry. Successful pilot programs can become models for practices to be imposed on subsequent software acquisitions. A number of advanced programming methodologies, including structured programming, top-down-design, formal specification languages, and automated validation methods are being developed to reduce the costs and improve the quality of software. These methodologies are prime candidates for selected DoD organizations to conduct pilot studies supported by the necessary resources, and to assess the results under realistic conditions. Some of the questions to be answered are:

- What are the costs of retraining programmers?
- What are the impediments to using these methodologies?
- Will these new techniques and tools require reorientation of existing organizational structures?
- Are there cheaper methods to achieve comparable results?

Activities now current in DoD acquisition programs which are applying these methodologies, for example pilot-structured programming projects in the Air Force, should be coordinated and supported at OSD levels.

3.3.7 Determine Realistic Weapon Systems Software Documentation Requirements

Current software documentation requirements should be examined to determine their utility to both the users and the developers of the system. Recommended changes should be incorporated into DoD regulations and standards for weapon systems software acquisition. Software documentation is described and required in numerous DoD regulations, standards and manuals (e.g., MIL-STD-483, MIL-STD-490, MIL-STD-881, MIL-STD-1521, DODM 4120.17M) and contract data requirements are described in DoD TD-3 'Authorized Data List'. It is not clear that all the documents are used or are useful. It is clear that they are expensive to produce and that descriptions are overlapping and sometimes contradictory. The Services and industry are attempting to consolidate documentation. A review is needed of the various audiences for these documents, and of their utility in use. The need for documentation for software projects with different functional type, complexity, and for different management methods should be assessed. The data item descriptions in TD-3 should be consolidated and duplications and conflicts resolved. Samples should be prepared for acquisition office use. Revised document types should be evaluted through trial program office applications.

3.3.8 Investigate Methods for Improving Software Testing Costs

A study should be conducted by DoD to determine the major elements of software testing costs (e.g., flight tests), and to determine methods to reduce them. Testing software for weapons systems under realistic conditions is very expensive. It involves flight tests of aircraft, test firings of missiles, and assembling large staffs to man command positions and evaluate test results. Test requirements and test methods should be investigated to determine ways of consolidating tests and of performing all but final tests without large expenditures. An investigation should be made of the just how much testing costs and where these costs occur. Methods of reducing these costs, such as consolidating testing and use of simulation techniques, should be investigated

3.3.9 Investigate Firmware Trends and Define DoD Firmware Policy Guidelines

This activity involves the investigation of ways in which policies and regulations should be revised to address firmware. Current acquisition policy does not address firmware or microprogramming. Policies and standards should be reviewed in light of the special characteristics of firmware - in some ways it is like software and in others like hardware. Revised or new regulations and standards should be prepared which specially address firmware acquisition. Recommendations should be made for the appropriate place for firmware in specifications, reviews and contractual documents.

3.3.10 Investigate More Effective Methods for Selecting Mutually Supportive Computer Hardware and Software Architectures

Research studies and experiments should be continued by the Services to determine principles for selecting computer hardware and software architectures which are mutually supportive and cost effective in meeting functional and performance requirements. This action can decrease software costs by providing a more realistic match between software requirements and hardware architecture and capabilities. The design experience of past systems should be reviewed where hardware/software tradeoff studies were conducted which resulted in unique software and hardware architectural designs. Systems should also be reviewed to determine the criteria which should be used to determine software/hardware architecture (e.g., flexibility required to meet changing requirements, and emphasis on use of available software and hardware). Future trends in computer hardware and software architectures are also important considerations.

3.3.11 Develop Techniques and Methods for Improving the Transfer of Technological Developments and Successful Practices Across Systems and Services

Efforts should be made to investigate and establish methods for improving the transfer of successful software practices across systems and Services, and for the transfer of technological developments from the laboratories or pilot programs to actual use in software acquisitions. The following types of actions should be included in a positive program of management and technology transfer:

For transferring research results to acquisition programs:

- Prepare and disseminate to acquisition offices summaries and assessments of software research, with identification of potential application areas for the results. These summaries should be prepared by an organization not associated with the research efforts.

- Stress policy and funding emphasis on evaluation and demonstration of software, i.e., on applied research and trial use.

- Establish personnel assignment policies which favor temporary assignment of technologists to acquisition offices and vice versa.

<u>For transferring information among programs</u>:

- OSD should fund, and encourage services to fund, independent reviews of programs to extract 'lessons-learned'. Separate organizations should prepare these reviews and OSD should be responsible for disseminating their results.

4. IMPLEMENTATION OF CORRECTIVE ACTIONS DURING PHASE II OF STUDY

INTRODUCTION TO SECTION 4

This section discusses the efforts that will be needed during the second phase of the DoD Software Steering Committee study program to further investigate and, where appropriate, to initiate implementation of the corrective actions recommended in Section 3. The discussion is organized into three topic areas.

1. Assumed Nature of Effort at Each DoD Management Level (Section 4.1)

It is suggested that the DoD Steering Committee, in assigning responsibility for further investigation and implementation of the corrective actions, assume three different levels of software acquisition management responsibility. The three levels are OSD level, Service/Development Command level, and Project Office level. The appropriate roles and responsibilities at each level relative to implementing the corrective actions are discussed.

2. Recommended Phase II Task Organization (Section 4.2)

Several of the corrective actions can be initiated immediately as interim solutions; others will require further definition. Also, several of the corrective actions should logically precede others for most effective implementation. This subsection identifies the interdependencies and organizes the corrective actions (tasks) in the form of a Phase II worklist. An implementation chart is provided which relates the required tasks to the different corrective actions and to suggested time-phased products.

3. Long-Term Action (Section 4.3)

The recommended actions of this report and the initial implementation of these and/or other actions during Phase II of the study must be followed-up by a long-term action program. This section discusses the limitations of ad hoc groups in effecting long-term solutions.

4.1 Assumed Nature of Effort at Each DoD Management Level

The implementation of the corrective actions during Phase II of the study will require the cooperative efforts of the OSD and the Services. Specific organizational and fiscal resources must be identified and lead responsibilities assigned. It is suggested that the DoD Steering Committee, in assigning responsibility for further investigation and implementation of the corrective actions, assume the following three different levels of management responsibility.

OSD Level

- Provides focal point for implementation of Phase II study efforts.
- Initiates and coordinates the preparation, correction, and review of DoD-wide publications and action memoranda.
- Establishes Service Executive Agents for tri-Service working groups.
- Coordinates task efforts across the Services.

Service/Command Levels

- Provides support to the OSD focal point and tri-Service working groups.
- Coordinates the preparation, correction, and review of Service-wide publications and action memoranda.
- Implements studies, pilot programs, and research under OSD guidance.
- Provides knowledgeable personnel from Service centers of expertise to support Phase II activities.

Project Office Level

- Provides inputs to the preparation of guidelines and new practices in the form of project experiences and 'lesson learned'.
- Supports studies required to identify new policies and practices.
- Supports pilot programs required to evaluate new acquisition guidelines and practices.
- Applies new guidelines and policies to programs and provides feedback of results.

4.2 Recommended Phase II Task Organization

For the purpose of ordering the Phase II activities, the recommended corrective actions can be grouped into the following tasks:

Those Tasks Which Can be Implemented Immediately as Interim Solutions

- Initiate OSD action to require specific DCP software support documentation for major systems. Require similar documentation at Service levels for non-major systems.

- Initiate OSD action to require the collection and dissemination of selected weapon system software management information including software-related cost, performance, and schedule information.

- Assign/allocate resources to review the major DoD publications used for software acquisition in order to identify interim corrections to major software deficiencies, inconsistencies, and conflicts.

Those Tasks Requiring Further Investigation and Definition Leading to Formal Longer-Term Solutions

- Assign resources to update and to expand the current series of DoD Directives, Instructions, and Standards to include comprehensive and consistent software performance specification and acquisition planning terminology, measures, policies, and guidelines.

- Develop task statements for the recommended software technology studies, pilot programs, and research areas. Allocate funds and assign responsibility to initiate, support, and coordinate the results of these programs.

- Initiate a formal investigation of software personnel practices and develop recommendations for improving them.

Table 4-1 provides an overview of a suggested implementation of the Phase I recommended corrective actions. It maps the corrective actions discussed in Section 3 to the above task groupings

TABLE 4-1
OVERVIEW OF SUGGESTED IMPLEMENTATION OF PHASE II CORRECTIVE ACTIONS

	Possible Products from Phase II Activities		
Corrective Actions (Sections Discussed)	Near Term 2 to 3 Mo.s	Intermediate Term 6 to 9 mo.s	Long Term Greater Than 12 mo.s
● Initiate Interim OSD/Service Policies			
- Require Submission of DCP Supporting Documentation (3.1.4) - Require Collection of Selected Management Information (3.2.6)	● Prepare Letter Directives for OSD Action		● Supercede Letter Directive(s) When Formal DoDD, DoDI Changes Issued.
- Review Major Software Documents to Identify Interim Improvements (3.2.7)	● Assign/Allocate Resources; Initiate Review.	● Complete Review of Major DoD Publications Used in Acquisition of Software; Prepare Interim Changes to Cover Major Deficiencies; Initiate OSD/Service Review.	● Supercede Interim Changes When Formal DoDD, DoDI, MIL Standards Issued.
● Initiate Longer Term Actions Leading to Formal Definition of DoD Software Directives, Instructions, MIL Standards			
- Define Major Software Performance Specification Factors (3.1.1) - Define Supporting Studies and Analyses (3.1.2) - Establish Formal Software QA Practices (3.1.3) - Define Major Software Acquisition Planning Factors (3.2.1) - Define Software Phases and Milestones (3.2.2) - Define Software Acquisition/Procurement Strategies (3.2.3) - Define Software Terminology (3.2.4) - Define Software Catalogue Requirements (3.2.5)	● Prepare Task Statements Assign/Allocate Resources; Initiate Tasks.	● Provide Recommended Policy Changes for Each Task Area Initiated During Phase II for Use as Inputs to Preparation of Formal, Long-Term Policy Improvements.	● Preparation of Draft of Formal, Long-Term Software Acquisition Policy Improvements; Includes Changes to DoDD, DoDI, and MIL Standards and New Publications if Appropriate; Initiate OSD/Service Review.
● Initiate (and Encourage Existing) High Leverage Studies, Pilot Programs, Research			
- Develop Quantitative Software Engineering Measures (3.3.1) - Improve 'Portability' of Software (3.3.2) - Investigate Automatic Programming Methods (3.3.3) - Define Methods for Predicting Cost of Software (3.3.4) - Investigate Areas and Methods for Effective Software Standardization (3.3.5) - Establish Pilot Programs Using Latest Methodologies (3.3.6) - Define Realistic Documentation Requirements (3.3.7) - Investigate Methods for Improving T&E Costs and Performance (3.3.8) - Investigate Firmware Trends; Define Policies (3.3.9) - Investigate Hardware/Software Architectures (3.3.10) - Transfer of Good Software Practices (3.3.11)	● Identify Existing Related DoD/Government/Industry Activities; Prepare Task Statements for New Areas; Assign/Allocate Resources; Initiate New Tasks.	● Establish Methods for Disseminating Research Results and Successful Practices Across Projects and Across Services. ● Disseminate Results from On-Going Research Activities to Services.	● Disseminate Initial Results to Services from Activities/Tasks Initiated During Phase II.
● Personnel			
- Investigate Personnel Practices (3.4)	● Establish Service Review Group	● Submit Recommendations to DoD Management	

and identifies the general nature of the Phase II task products and their relative time frame.[1]

4.3 Long-Term Action

Studies aimed at improving the acquisition process seem inevitably to be conducted by ad hoc groups. But no matter how valid their results, they are not likely to bring about significant long-term improvement. This is because the improvement problem is essentially not one of a lack of knowledge which, once the needed knowledge is supplied, causes the problem to go away. Rather, resolving the problem is a matter of providing for:

1. Continuous gathering of the "lessons learned" from actual experience in the complex and uncertain undertakings systems acquisitions represent.

2. Continuous analysis of, and a "corporate memory" related to, this experience.

3. Continuous widespread education of a type that reflects the high turnover of both military and higher-level civilian personnel in DoD, in how to apply the lessons we have learned from experience to a particular case (particularly what should be taken into account in devising the acquisition strategy for the particular case).

4. Regular staffs following up on the results of ad hoc study groups to assure both that the findings of the groups are valid when applied to real cases and, if valid, that they are applied to other cases.

5. And finally, a regular mechanism for both translating the findings of an ad hoc group -- when in fact it arrives at some valuable new knowledge or intuition -- into preliminary policy direction, and for conducting the subsequent controlled experiment by which such preliminary policy direction can become fully accepted.

In summary, there is a need for a centralized staff resource which is tasked with the long-term responsibility for coordinating and providing follow-up on major software acquisition issues across all of the DoD.

[1] Examples of material that may be useful in developing the Phase II products are included in Appendices G and H.

APPENDIX A

STUDY DEFINITIONS

(Common Terms Used in Report)

Firmware -- Firmware differs from software in that it is not easily alterable such as for software. Use of read-only memories (ROMs) or programable read-only memories (PROMs) in processors or special purpose hardware are examples of firmware.

Software -- The term software is used to refer to computer programs, associated data bases, and related documentation required to define, design, develop, produce, test, operate, and maintain the software-related aspects of the total weapon system, including computer hardware, software, personnel and procedures.

Software Acquisition -- The term software acquisition as used in this report refers to all life cycle phases including early concept definition and validation through to operations and maintenance after systems transition. It also includes major software developments involved in updating or improving already operational systems.

Software Element or Component -- A software element or component within a weapon system refers to the major groupings of software within that system. For example: operational software, automatic test equipment software, maintenance support software, etc.

Software Life Cycles Phases -- Refers to all phases from concept definition and validation through to operation and maintenance including major updates. (See software acquisition above.)

Software Operations and Maintenance (O&M) -- Refers to all software-related activities concerned with the ongoing operation and maintenance (ownership) of software integral to the weapon system and in supporting subsystems. It generally occurs after system transition.

Software Subelement or Subcomponent -- Refers to the next level breakdown below an element or component. For example: operational software packages for specific computers within a total weapon system or separate ATE diagnostic packages.

Validation -- The term validation alone is related to the testing and evaluation process and is used in a general sense in this report. For example, validation can refer to concept validation, contractor software validation, or validation facilities.

Verification and Validation (V&V) -- V&V refers to the specific use of V&V personnel and facilities to validate/verify that a weapon system's performance meets both specified contractor requirements and user mission requirements.

APPENDIX B

OSD MEMORANDUM INITIATING STUDY

OFFICE OF THE SECRETARY OF DEFENSE
WASHINGTON, D.C. 20301

3 DEC 1974

MEMORANDUM FOR The Assistant Secretary of the Army
(Installations and Logistics)
The Assistant Secretary of the Navy
(Installations and Logistics)
The Assistant Secretary of the Air Force
(Installations and Logistics)
The Assistant Secretary of the Army
(Research and Development)
The Assistant Secretary of the Navy
(Research and Development)
The Assistant Secretary of the Air Force
(Research and Development)

SUBJECT: Management of Weapon System Software

The sharply rising costs of software programs in the weapon system acquisition process, with respect to acquisition procedures, development and maintenance of such software, and the increasing importance of the software role in the overall mission effectiveness of major DoD weapon systems constitute serious technical and management problems that must be solved if we are to have the weapon systems that are needed for our national security. To find solutions to these problems, we are initiating a two phase study program which will require the joint involvement of the OSD staff and the Services.

The first phase of the study program is only now starting. Its major effort centers on two four month studies by the Mitre Corporation and the Applied Physics Laboratory at Johns Hopkins University to identify and define (1) the nature of the critical software problems facing the DoD, (2) the principal factors contributing to the problems, (3) the high pay-off areas and alternatives available, and (4) the management instruments and policies that are needed to define and bound the functions, responsibilities and mission areas of weapon systems software management. The second phase of the study program will be to examine in depth those areas which have been surfaced in the first phase as having first-order importance to the DoD. It is not

OFFICE OF THE SECRETARY OF DEFENSE
WASHINGTON, D.C. 20301

3 DEC 1974

MEMORANDUM FOR The Assistant Secretary of the Army
(Installations and Logistics)
The Assistant Secretary of the Navy
(Installations and Logistics)
The Assistant Secretary of the Air Force
(Installations and Logistics)
The Assistant Secretary of the Army
(Research and Development)
The Assistant Secretary of the Navy
(Research and Development)
The Assistant Secretary of the Air Force
(Research and Development)

SUBJECT: Management of Weapon System Software

The sharply rising costs of software programs in the weapon system acquisition process, with respect to acquisition procedures, development and maintenance of such software, and the increasing importance of the software role in the overall mission effectiveness of major DoD weapon systems constitute serious technical and management problems that must be solved if we are to have the weapon systems that are needed for our national security. To find solutions to these problems, we are initiating a two phase study program which will require the joint involvement of the OSD staff and the Services.

The first phase of the study program is only now starting. Its major effort centers on two four month studies by the Mitre Corporation and the Applied Physics Laboratory at Johns Hopkins University to identify and define (1) the nature of the critical software problems facing the DoD, (2) the principal factors contributing to the problems, (3) the high pay-off areas and alternatives available, and (4) the management instruments and policies that are needed to define and bound the functions, responsibilities and mission areas of weapon systems software management. The second phase of the study program will be to examine in depth those areas which have been surfaced in the first phase as having first-order importance to the DoD. It is not unlikely that a study group will be organized at this time having the following objectives: Identify and evaluate current and alternative Defense and commercial software policies and practices in development, procurement and operational support which most significantly influence acquisition and life cycle costs, field reliability, maintenance, standardization and to identify possible improvements to reduce and control costs and improve software reliability, standardization, maintainability and software research and development production capabilities.

The software study program needs direct service participation by military officers or civilian experts experienced in requirements generation, weapon systems acquisition, support and management techniques as they apply to software. In the first phase of the effort these needs can best be met by having two individuals from each Service identified to serve on the Software Steering Committee. It is recommended that one individual have an R&D background and the other have logistics experience. It is not anticipated that the services of the individuals identified will be required on a full time basis.

The Committee will assist in developing the study goals for each phase of the total effort, provide focal points within the DoD to coordinate and support the study objectives, assist in obtaining the data needed in accomplishing the studies and to make recommendations on how to implement study findings and to determine the nature and extent of the follow-on activities. It is suggested that personnel at the 0-6 level be considered for assignment to the Committee and that they be selected on the basis of the Committee's needs, responsibilities and objectives as outlined above.

The first meeting of the Software Steering Committee, with the contractors, is planned for Friday, 13 December at 1330 hours in Conference Room 1E 801 #4. You are requested to have the names of your committee representatives to Col. R. D. Hensley, OASD(I&L)WA, Room 2A 318 prior to this date.

MALCOLM R. CURRIE
Director of Defense
Research and Engineering

ARTHUR I. MENDOLIA
Assistant Secretary of Defense
(Installations and Logistics)

Terence E M'Clary

TERENCE E. McCLARY
Assistant Secretary of Defense (Comptroller)

APPENDIX C

BIBLIOGRAPHY
STUDY DOCUMENTATION BASELINE

1. CCIP-85. Information Processing/Data Automation Implications of Air Force Command and Control Requirements in the 1980's: Executive Summary, Revised Edition. Los Angeles, California; Air Force Systems Command, Space and Missile Systems Organization, February 1972. (AD 742 292)

CCIP-85. Information Processing/Data Automation Implications of Air Force Command and Control Requirements in the 1980's: Eleven Volumes. Los Angeles, California; Air Force Systems Command, Space and Missile Systems Organization.

Volume I	Highlights	(AD 900 031L)
Volume II	Command and Control Requirements: Overview	(AD 521 887L)
Volume III	Command and Control Requirements: Intelligence	(AD 523 881L)
Volume IV	Technology Trends: Software	(AD 919 367L)
Volume V	Technology Trends: Hardware	(AD 907 626)
Volume VI	Technology Trends: Sensors	(AD 525 661)
Volume VII	Technology Trends: Integrated Design	(AD 906 757L)
Volume VIII	Interservice Coordination Trends	(AD 522 216L)
Volume IX	Analysis	(AD 524 549)
Volume X	Current Research and Development	(AD 905 654L)
Volume XI	Integrated Research and Development Roadmaps	(AD 902 515L)

2. Electronics-X: A Study of Military Electronics with Particular Reference to Cost and Reliability. Volume 2: Complete Report. Arlington, Virginia; Institute for Defense Analyses, Science and Technology Division; January 1974. (R-195) (Volume 1 was not available for referencing in this bibliography.)

3. Fisher, David A., Automatic Data Processing in the Defense Department. Arlington, Virginia; Institute for Defense Analyses, Science and Technology Division; October 1974. (P-1046) (Contract DAHC15-C-0200, Task T-36)

4. Goldberg, Jack, ed. The High Cost of Software, (Proceedings of a Symposium). Sponsored by: Air Force Office of Scientific Research, Army Research Office, and Office of Naval Research, Monterey, California; 17-19 September 1973. Published: Menlo Park, California; Stanford Research Institute, n.d. (Contract N00014-74-C-0028)

5. Government/Industry Software Sizing and Costing Workshop, (Summary Notes). L. G. Hanscom Field, Bedford, Massachusetts; Air Force Systems Command, Electronic Systems Division, 1-2 October 1974. (Draft)

6. Manley, John H., Colonel, and Archibald, W. Robert, eds. Aeronautical Systems Software Workshop, (Proceedings). Sponsored by: Headquarters, Air Force Systems Command, Aeronautical Systems Division, Wright-Patterson Air Force Base, Dayton, Ohio, 2-4 April 1974. (Draft)

7. Project PACER FLASH. Four Volumes. Wright-Patterson Air Force Base, Dayton, Ohio; Air Force Logistics Command, 28 September 1973.

Volume I	Executive Summary and Final Report
Volume II	Appendix A: Automatic Test Equipment (ATE)
Volume III	Appendix B: Operational Flight Program
Volume IV	Appendix C: Air Crew Trainers (Simulators)

8. Reich, Eli T. Tactical Computer Software Acquisition and Maintenance Staff Study. Washington, D. C.; Deputy Assistant Secretary of Defense, Production Engineering and Materiel Acquisition, 31 October 1973.

9. A Report on Air Force Logistics. Two Volumes. Command Operation Operation`Flight Program Support. Santa Monica, California; System Development Corporation, 9 December 1974. (TM-5439/000/00)

Volume I	Introduction and Summary (TM-5439/000/00)
Volume II	Data Analysis and Conclusions (TM-5439/001/00)

10. Report of Army Scientific Advisory Panel Ad Hoc Group on Army Tactical Data System Software Development. October 1974.

11. Bernard A. Zempolich. An Analysis of Computer Software Management Requirements for Operationally Deployable Systems, Executive Summary (Volume I). Department of the Navy.

APPENDIX D

WEAPON SYSTEMS INTERVIEWS

There were five systems of the Army and nine of the Air Force reviewed during this study. In addition, the Joint Integration Test Facility (JITF) at San Diego was visited. The JITF is responsible for joint interoperability of Service-developed systems performing mission roles of tactical air control and tactical air defense (TACS/TADS). The systems reviewed are as follows:

Army	Air Force		JITF
TACFIRE	DSP	485L	TACS/TADS
Q-73	MINUTEMAN	427M	
PERSHING	F-111	COMBAT GRANDE	
SAM-D	WILD WEASEL	AWACS	
SAFEGUARD	B-1		

The reviews were conducted with software management and engineering personnel of the Army and Air Force responsible for managing the development, testing and transitioning of the software. Prior to each visit, a questionnaire was sent to the system management offices to provide an indication of the information required for the study. The personnel interviewed were very cooperative and courteous with the MITRE Team.

The systems reviewed cut across all stages of the acquisition process. Table D-1 indicates the systems reviewed, their status in the acquisition process, and the organizations and dates of the interviews. A detailed discussion of the results of these interviews is provided as separate material in Volume II of this report.

TABLE D-1
OVERVIEW OF SYSTEMS REVIEWED

	NAME	MISSION/ROLE	STATUS	INTERVIEWS
AVIONICS				
	B1	STRATEGIC BOMBER	DEVELOPMENT	ASD SPO, MAR 7.
	WILD WEASEL	AIRBORNE SAM DEFENSE	LIMITED PRODUCTION/UPDATE	ASD SPO, MAR 7.
	F-111 (B,D,F)	STRATEGIC/TACTICAL BOMBER	DEPLOYED/UPDATE	SACRAMENTO ALC, FEB 27.
MISSILES & GROUND SUPPORT				
	SAFEGUARD	MISSILE DEFENSE	TRANSITIONING	BMDPO, JAN 9; BMDSC, FEB 7.
	PERSHING	TACTICAL MISSILE	DEPLOYED/UPDATE	MISSILE COMMAND PMO, FEB 6.
	SAM-D	AIR DEFENSE	DEVELOPMENT	MISSILE COMMAND PMO, FEB 6.
	MINUTEMAN	STRATEGIC MISSILE	DEPLOYED/UPDATE	SAMSO SPO, FEB 25.
CONTROL				
	TACFIRE	ARTILLERY FIRE SUPPORT	LIMITED PRODUCTION	ARTADS PMO, FEB 4.
	AN/TSQ-73	MISSILE MINDER-AIR DEFENSE	LIMITED PRODUCTION	ARTADS PMO, FEB 7.
	DSP	INTELLIGENCE	DEPLOYED/UPDATE	SAMSO SPO, FEB 24.
	485L	TACTICAL AIR CONTROL	DEVELOPMENT	ESD SPO, MAR 12.
	427M	CONTINENTAL AIR DEFENSE, SPACE SURVEILLANCE	DEVELOPMENT	ESD SPO, MAR 12.
	COMBAT GRANDE	AIR DEFENSE FOR SPAIN	DEVELOPMENT	ESD SPO, MAR 13.
	AWACS	AIRBORNE TACTICAL C&C	PRODUCTION PENDING	ESD SPO, MAR 13.
INTEROPERABILITY				
	TACS/TADS	INTEROPERABILITY TESTING	TEST; PREPARING FOR OED	JITF, FEB 26.

APPENDIX E

STUDY PARTICIPANTS

The following is a list of the principal participants and persons interviewed during this study. Not all persons who attended various conferences and interviews are included in the listing to avoid the list from becoming too lengthy. A word of thanks, however, goes to all those who participated.

The MITRE Corporation Study Team

Mr. A. Asch
Mr. T. L. Connors
Mr. D. W. Kelliher
Mr. J. P. Locher, III

The MITRE Corporation Software Review Group and Workshop

Mr. W. S. Attridge
Mrs. J. A. Clapp
Mr. R. P. Foreman
Mrs. J. D. Greenwood
Mr. T. C. Hilinski
Mr. W. F. Potter
Dr. E. L. Rabben
Mr. E. Raichelson
Mrs. K. K. Rebibo
Mr. J. K. Summers
Dr. N. Waks

The DoD Software Steering Committee

OSD Members

Rear Admiral D. Webster, I&L
Col. R. D. Hensley (USAF), I&L
Lt. Col. W. A. Whitaker (USAF), ORDDR&E
Mr. W. Franklin, OASD (C)
Mr. C. R. Pack, OASD (C)

Army Members

Mr. D. I. Ciliax, U.S. Army Missile Command
Mr. F. W. Fairchild, AMCR-D

Navy Members

Rear Admiral R. S. Smith, OP-34
Capt. J. D. Elliott, OP-34B
Capt. J. E. Fernandes, PM-18T
Capt. E. J. Richer, MAT-09Y
Mr. H. Sonnemann, DASN (R&D)

Marine Corp Member

Maj. L. E. Obenhauss, RDD-23

Air Force Members

Col. J. P. Cooper, AF/RDM
Col. E. P. Eaton, AF/LG

Other Members

Dr. D. A. Fisher, IDA
Dr. J. C. R. Licklider, ARPA

Department of the Army Interviews

Army Commands

Mr. F. W. Fairchild, AMCR-D (Coordinator for all Army interviews)
Col. R. M. Ward, CSC
Mr. J. C. Domingue, CSC
Col. D. Lasher, CENTACS
Mr. A. Coleman, CENTACS
Dr. E. Lieblin, CENTACS
Mr. D. Ciliax, MICOM-RD (Coordinator for all interviews at Huntsville)

Army Weapon Systems Interviews

TACFIRE

Mr. N. Atkinson, ARTADS
Mr. N. Taupeka, ARTADS
Mr. N. Thompson, ARTADS

Safeguard

Dr. R. Mervin, BMDPO
Col. L. Heigert, BMDSC
Mr. D. R. McClung, BMDSC

Pershing

Mr. W. Wagner

SAM-D

Maj. E. R. Jackson
Mr. W. Mobley

TSQ-73

Capt. W. R. Lynn
Mr. B. Owen

Department of the Air Force Interviews

Air Force Headquarters and Commands

Col. J. P. Cooper, AF/RDM
Col. E. P. Eaton, AF/LG
Lt. Col. R. J. Vodicka, AF/RDM (Coordinator for Air Force interviews)
Lt. Col. J. H. Manley, AFSC
Lt. Col. T. Yamamoto, AFSC
Lt. Col. D. L. Butler, SAMSO (Coordinator for SAMSO weapon system interviews)
Maj. H. Falk, ASD
Mr. P. Johnson, ASD (Coordinator for ASD weapon system interviews)
Capt. W. White, ESD (Coordinator for ESD weapon system interviews)

Air Force Weapon Systems Interviews

DSP

Col. J. J. Mularz
Lt. Col. R. Lawrence

Minuteman

Lt. Col. J. L. Fisher
Maj. A. J. Driscoll
Capt. R. Gounaud

F-111

Mr. A. Patterson
Capt. T. O. Nickerson
Mr. D. Sturdeyant
Mr. J. Naley

Wild Weasel

Lt. Col. M. Bradley
Maj. J. Logan
Mr. B. Vanglin
Mr. J. Turner
Lt. R. Mundi (AFLC Liaison)

B-1

Maj. Gen. A. B. Martin
Lt. Col. J. J. Canaday
Mr. D. Holtz
Mr. H. Peat

Combat Grande

Maj. H. E. Routh, Jr.
Capt. G. P. Walsh
Mr. S. Pomponi (MITRE)
Mr. F. D. O'Connor (MITRE)

485L

Mr. G. M. Sheenfeld
Mr. N. E. Bolen (MITRE)

427M

Lt. Col. R. I. Roseman
Lt. Col. H. E. Carolus

AWACS

Maj. E. E. Gould
Capt. E. J. Morrison

Other Interviews

Mr. H. P. Gates, Economic Scientific Planning
Mr. W. Dowden, SDC
Mr. H. P. Dowst, SDC
Mr. D. Vandaveer, SDC
Mr. B. Zempolich, Department of the Navy
Mr. R. A. Eidson, Decisions and Design, Inc.
Capt. D. F. Cross (USN), JITF Director TACS/TADS
Col. D. E. McPherson, Jr. (USAF), AF Director TACS/TADS

APPENDIX F

MANAGEMENT WEAPON SYSTEM SOFTWARE COST MODEL

1. BACKGROUND/PURPOSE

Three major difficulties confront management in attempting to estimate annual costs of weapon systems software in the DoD. They are:

1. No formal agreement on which systems fall under the weapon systems classification.

2. No formal agreement on which costs should be attributed to software in weapon systems.

3. Lack of meaningful cost data for most weapon systems since no formal definitions have existed in the past and no detailed cost records were kept.

The primary purpose of the management cost model developed in this appendix is to provide a starting point for resolving 1 and 2 above, that is, to provide a preliminary definition of which systems fall under the weapon systems classification and to provide management with a 'strawman' list of factors contributing to the cost of software in these weapon systems.

A secondary purpose is, through the use of 'typical' costs, to arrive at annual software cost estimates. However, the room for error and misinterpretation is very large in this approach, and until such time as the model can be widely reviewed, refined, and more accurate cost data points obtained from the actual system managers, any results obtained should be viewed and used cautiously.

The model should also be viewed as a simplistic rather than complex model. There is a tendency to provide detail in the hope of achieving accuracy. However, with too much detail the model would lose comprehension. In this initial model, we have attempted to stay at a fairly broad level.

2. ASSUMPTIONS

2.1 Weapon Systems Definition

The systems on the 'SAR Coverage by Weapon System' list, dated

September 30, 1974, represent an initial definition of which systems are in the weapon systems classification. (There are 79 systems in this list: 42 under congressional review; 8 under current reports for OSD, GAO, OMB; and 29 under discontinued reporting.) To this list should be added known omissions such as 485L/407L (Air Force), Pershing I&II, and Q-73 (Army). Non-tactical command, control, and communications (C^3), intelligence, logistics, and automatic data processing (ADP) not associated directly with weapon systems design, development, or operations and maintenance (O&M) should be omitted. An initial weapon systems list is provided as Table F-1 in this appendix. The model should also recognize that some minor or older systems do have software associated costs and are not generally reflected in OSD and Service level weapon systems lists.

2.2 Model Assumptions

For the purpose of developing the model, weapon systems software costs are assumed to consist of the following five components:

2.2.1 Software Development Costs Associated with the Design, Development, Production, or Major Updates of Systems -- Includes:

- DoD software management costs (including in-house management personnel, separate consultants, validation contractors, etc.).

- Operational software development costs (including in-house and contractor costs to specify, design, develop, and test operational software -- including development tools (e.g., compilers, utility system), operating systems, application software, testware).

- ADP and scientific data processing associated with weapon systems.

- Separate government and contractor Validation and Verification (V&V) during development (facilities/personnel).

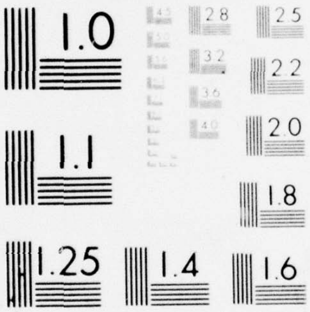
- Costs to develop operational support tools (including support software, Automatic Test Equipment (ATE) and diagnostics, Air Crew/Simulators).

- Portion of costs associated with firmware and use of software in hardware design, development, and production activities.

AD-A034 802 MITRE CORP MCLEAN VA F/G 9/2
DOD WEAPON SYSTEMS SOFTWARE ACQUISITION AND MANAGEMENT STUDY. V--ETC(U)
MAY 75 A ASCH, D W KELLIHER, J P LOCHER
UNCLASSIFIED MTR-6908-VOL-1 NL

2 OF 2
AD
A034802

END
DATE
FILMED
2-77



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963

- Software cost of hardware (includes contractor software normally provided as part of computer hardware).

2.2.2 Software Costs Associated with System Operations and Maintenance -- Includes:

- Cost to develop software maintenance tools and documentation not provided under development/production costs (e.g., data reduction and analysis, support software).

- Cost to improve, maintain operational software (e.g., operational flight programs).

- Cost to improve, maintain air crew training/simulator software.

- ADP support to items above.

- Software validation facilities (amortized over life of system).

- Portion of engineering flight testing required to validate software.

- Portion of user flight testing required to validate software.

- User software maintenance and overhead costs where separate.

2.2.3 Software Costs Associated with Separate System Test and Evaluation -- Includes:

- DT&E (Development Test and Evaluation).

- OT&E (operational Test and Evaluation).

- Interoperability Testing (software portion allocated to system).

- Combined Operations/Demonstrations (software portion).

2.2.4 Overhead Costs Associated with Software But Not Directly Charged to a Single System -- Includes:

- Studies, Pilot Programs.

- R&D (Software related technology areas).
- Multi-system T&E (e.g., JITF, TACS/TADS).
- OSD and Service Level Management Overhead.

2.2.5 Indirect Costs Attributable to Software -- Includes:

- Program Delays.
- Loss of Mission Effectiveness.
- Shorter Mission Life.

2.3 Weapon Systems Categories

For the purpose of developing the model, weapon systems are assumed to be categorized as follows:

1. Avionics and ground support.
2. Missiles and ground support/control.
3. Tactical control and information systems.
4. Shipboard systems.

For each category of weapon systems, three levels of software are involved: low -- limited to only minor support or operational functions; medium -- significant software perhaps in ATE or in another support function; and high -- large software elements involved in the system.

2.4 Cost Assumption

For the purpose of developing the model, it is assumed that typical coverage costs for each of the above weapon systems categories and the software level in a given category remain constant during the development and maintenance phases.

3. PROPOSED MODEL

3.1 Parameters

	Parameter	Section Assumption Explained
1.	Four Categories of Weapon Systems	2.3
2.	Twelve Levels of Software in Those 4 Categories and Software	2.3
3.	Number of Systems of Each Weapon Systems Category and Software Level in Development or Major Update Cycle	2.3
4.	Number of Systems of Each Weapon System Category and Software Level in O&M Phase	2.3
5.	Typical Annual Software Costs for Development for Each Category/Level	2.2.1
6.	Typical Annual Software Costs for O&M for Each Category/Level	2.2.2 2.4
7.	Typical Annual Software Costs for Separate T&E During Development for Each Category/Level	2.2.3 2.4
8.	Typical Annual Software Costs for Separate T&E During O&M for Each Category/Level	2.2.3, 2.4
9.	Indirect Cost Coefficient During Development	2.2.5
10.	Indirect Cost Coefficient During O&M	2.2.5
11.	Other Minor or Older System Coefficient During Development	2.1
12.	Other Minor or Older System Coefficient During O&M	2.1
13.	Annual Software Overhead Costs not Charged to Individual Systems.	2.2.4

3.2 Model (Annual Weapon System Software Cost Estimate)

$$\sum_{2}\left\{\left[\textcircled{3}\left(\textcircled{5}+\textcircled{7}\right)\textcircled{9}\textcircled{11}\right] + \left[\textcircled{4}\left(\textcircled{6}+\textcircled{8}\right)\textcircled{10}\textcircled{12}\right]\right\} + \textcircled{13}$$

4. BACK-OF-ENVELOPE ANNUAL WEAPON SYSTEMS SOFTWARE DIRECT COST ESTIMATE

MITRE was requested as part of this study to obtain a gross estimate of total weapon systems software costs. Attempts to obtain sufficient data to make a defensible estimate were not successful; however, we have tried to make an educated guess based upon this cost model and eight-to-ten data points obtained during the study. The results obtained are probably over qualified and over conservative.

4.1 Approach

Two methods were addressed:

1. Make estimates for annual costs in typical weapon systems and then multiply by the number of weapon systems.

2. To check whether these estimates are in an acceptable range, estimate the size and cost of the total DoD weapon system in-house and out-of-house software development and production capability.

4.2 Estimate Using Typical Costs for Typical Systems

As previously noted, three major difficulties exist in attempting to estimate weapon systems software costs:

1. No formal agreement on which DoD systems fall under the weapon system classification.

2. No formal agreement on which costs should be attributed to software in weapon systems.

3. Lack of meaningful cost data since a separate breakout for software hasn't been consistently kept in the past.

The first item above is best resolved by listing the specific systems being costed. Table F-1 is such a list. It starts with the 'SAR Coverage by Weapon System' list, dated 30 September 1974, obtained from OASD (I&L). To it are added systems that appear to be missing. Generally excluded are non-tactical C^3, early warning systems, logistic, and ADP systems. Table F-1 is a 'mixed bag' at best, but it is a starting point.

The second item (which costs are attributable to software) is resolved by developing a cost model of which costs are included.

TABLE F-1
LIST OF WEAPON SYSTEMS

(Time did not allow verification that the list is complete, truly representative of accepted weapon system definitions, nor that correct designations were used.)

I. Systems Listed on 'SAR Coverage by Weapon System', 9/30/74.

Army

- LANCE
- Imp. HAWK
- SAFEGUARD
- SAM-D
- HLH
- UTTAS
- MICV
- ARSV(SCOUT)
- AAH
- XM-1
- DRAGON
- TOW
- TACFIRE
- MANPADS (STINGER)
- AN/TTC-39
- SHERIDAN
- Ammo. Annex
- SHILLELAGH
- GAMA GOAT
- MBT-XM803
- CHEYENNE
- M60A1E2

Navy

- A-7E
- E-2C
- F-14A
- P-3C
- S-3A
- AEGIS
- CONDOR
- HARPOON
- PHOENIX
- POSEIDON
- SIDEWINDER AIM-9L
- SHRIKE
- STANDARD ARM
- WALLEYE II
- AN/SQS-26
- DIFAR
- DSRV
- VAST-335
- SFARROW III E
- SSN
- DE-1052

- SPARROW III F
- TRIDENT
- MK-48
- SSN-688
- DD-963
- DLGN-38
- LHA
- CVAN 68 Class
- PF
- PHM

- SSN-685
- AN/SQQ-23
- DLG AAW MOD
- AMTRAC
- AV-8A
- AN/BQQ-5
- EA-6B

Air Force

- A-7D
- A-10
- B-1
- F-5E
- F-15
- F-111A/D/E/F
- AWACS
- AABNCP
- MAVERICK
- MINUTEMAN III, II
- SRAM
- RF-111
- FB-111A
- C-5A

II. Other Systems

- MSG4
- TOS
- ATMAC
- ADCCS
- SERGEANT
- HONEST JOHN
- CHAPARREL/VULCAN
- RED EYE
- PERSHING I, II
- Q-73

- TARTAR
- TALOS
- TERRIER
- SEA SPARROW
- MACCS
- MTACCS
- JPTDS
- IFDS
- AFDS
- ITAWDS
- F-8
- F-4
- EA-6A
- OV-10A
- A-4
- A-6
- CH-46A
- CH-53D

- NTDS
- NIPS
- OSIS
- MAGIS

- TITAN
- B52
- F-106
- F-101
- F-102
- EC-121
- NIKE-HERCULES
- 485L/407L
- TIPI

The cost model developed in this appendix is such a model. In general, the direct cost components of the model are used here.

For cost data, very liberal use is made of the eight or ten data points obtained during the study interviews.

Table F-2 is a compilation of the above, based on the 115 systems listed in Table F-1. The distribution between types of systems (40%, 35%, 25%) and the number in development or update versus operations and maintenance (O&M) are based on a rough sample of the Table F-1 systems (some are in both). The typical cost ranges are educated guesses in most instances, but are still probably on the conservative (low) side when all hidden government costs are considered.

The system-related (non-common) weapon systems software annual cost estimates from Table F-2 total $.558 to 1.396 Billion. To this figure should be added common software costs (e.g., R&D, TACS/TADS, laboratories) assumed to be on the order of $200 M. This addition gives a rough estimate for direct software-related weapon systems software costs of $.8 to 1.6 Billion annually. The uncertainties result in the large spread. The probability that the low figure of $.8 B is still high is almost zero; the probability that the $1.6 B isn't high enough depends on how honest one wants to be in collecting the hidden government costs and on whether one wants to add indirect costs (due to the impact of software on program delays, shorter mission life, or loss of weapon effectiveness).

4.3 Estimate of Total DoD Weapon System Software Development and Production Capability

After several false starts, the most sensible method of obtaining a total cost estimate is to use the method and results developed in CCIP-85 and refined in Dave Fisher's report, ADP Costs in the Defense Department, IDA Paper P-1046 dated October 1974. In the IDA report, the software cost estimate for FY73 for DoD systems unreported in the GSA inventory (assumed to be largely weapon systems) was $1.3 to 1.9 Billion.

Table 13 from the IDA report is attached, and includes the major cost factors and assumptions used. In general, the approach followed was to use FY73 line item budget figures with percentages for software derived from analogies with industrial firms doing similar work. While large errors are possible in using this method, the basic assumptions have been around since the CCIP-85 Report (with no better approach forthcoming to the author's knowledge).

TABLE F-2
BACK OF THE ENVELOPE WEAPON SYSTEM SOFTWARE ANNUAL COST GUESSTIMATE[1]

Type of System	'Amount' of Software[3]	Number of Systems	Number in Development	Estimated Range of Software Development $s Per System[2] (Millions)	Total Development $s Per Category	Number in O&M	Estimated Range of Software O&M $s Per System[2] (Millions)	Total O&M $s Per Category
● Information and Control (40%)								
	Hi	15	9	15-30	135-270	9	8-15	72-135
	Medium	15	9	5-15	45-135	9	4-8	36-72
	Lo	16	9	1-5	9-45	9	1-4	9-36
● Missiles & Deployed Weapons (35%)								
	Hi	13	8	10-25	80-200	8	8-15	64-120
	Medium	13	8	2-10	16-80	8	2-8	16-64
	Lo	14	8	.5-2	4-16	8	.5-2	4-16
● Avionics & Ground Support (25%)								
	Hi	9	5	5-15	25-75	5	5-12	25-60
	Medium	10	6	1-5	6-30	6	1-5	6-30
	Lo	10	6	.5-1	3-6	6	.5-1	3-6
Totals		115	68		$323-857M	68		$235-539M

[1] System oriented direct costs only; Indirect costs (e.g., program delays, loss of mission effectiveness) and non-system related costs (e.g., R&D) not included.

[2] Estimates include direct costs including contracts, government system-related overhead costs, test and evaluation (OT&E, DT&E, OED, etc.).

[3] Assumes all systems in Table A use some amount of support software.

TABLE 13
ESTIMATED DOD SOFTWARE COSTS, COMPUTER SYSTEMS UNREPORTED IN GSA INVENTORY, FY 1973
(dollars in millions)

	Air Force		Army		Navy		Other DOD		DOD Total	
Aircraft, Missile, & Ship Procurement	$4,326		$814		$6,635		--		$11,775	
Software @ 3.4%-5.5% of Above		$147-$238		$28-$45		$226-$365		--		$400-$648
Intelligence & C³	--[a]		--[a]		--[a]		--[a]		--[a]	
Software @ 8%-12% of Above		--		--		--		--		--
RDT&E										
Budget	$3,120		$1,885		$2,542		$446		$ 7,995	
Estimated In-House Personnel	1,000		605		816		143		2,566	
Total	$4,120		$2,490		$3,358		$589		$10,559	
Software @ 7%-10% of Above		$288-$412		$174-$249		$235-$336		$41-$59		$739-$1,056
SOFTWARE SUBTOTAL, ABOVE THREE ITEMS		$435-$650		$202-$294		$461-$701		$41-$59		$1,139-$1,704
Software Cost of Hardware @ 15% of Above		$65-$98		$30-$44		$69-$105		$6-$9		$171-$256
SOFTWARE TOTAL, SYSTEMS UNREPORTED IN GSA INVENTORY		$500-$748		$232-$338		$530-$806		$47-$68		$1,310-$1,960

[a] Value unknown.

Note: This table is a copy from IDA Paper P-1046, ADP Costs in the Defense Department, dated October 1974.

4.4 Summary

The above two cost estimates of $.8 to 1.6 Billion and $1.3 to 1.9 Billion are gross estimates at best. However, for DoD management purposes, it appears safe to assume they are in the correct range. The next level of refinement would be to meet directly with DoD personnel to obtain a more accurate Table F-1 (list of agreed upon weapon systems), a more accurate distribution across systems ('amount' of software and number in development versus deployment), and additional cost data points.

APPENDIX G

SAMPLE MATERIAL FOR USE IN PREPARING NEAR-TERM ACTIONS

(Sample Action Memorandum)

SUBJECT: Requirements for Additional Software Planning Information
in Support of Future OSD and Service Weapon System Reviews

It has been recommended by the DoD Weapon System Software Management Steering Group that specific software tradeoff analyses and planning documentation be required as supporting material for all weapon systems as part of the DCP/DSARC review process for major systems, and that similar requirements be levied at Service levels for other systems involved or about to be involved in significant software development activities. While adequate analyses and early software planning is performed in many instances across the Services, there is an apparent lack of consistency across all systems. With the increasing frequency of use and the increasing weapon systems mission dependence on software, appropriate action must be taken to insure that good software acquisition practices leading to the maximum value per DoD dollar spent are being followed. Towards this end, the following two actions are being initiated through this memorandum:

1. The DoD Weapon Systems Software Management Steering Group should identify specific OASD and Service resources required to develop a 5000 series regulation specifically oriented towards insuring that all software planning factors are adequately analyzed and reviewed in future OSD and Service review processes. A draft regulation should be made available for coordination and review within 120 days from receipt of this memorandum.

2. In the interim and until such time as the new regulation is adequately reviewed and approved, the following specific software planning documentation will be required in support of all OSD and Service system reviews:

a. A software acquisition management plan;

b. An analysis of new software developments and facilities versus use of existing resources;

c. An analysis of software development risks involved; and

d. An analysis of the proposed software design approach versus use of hardware or other design approaches.

A more detailed description of this supporting documentation is provided as a separate attachment. All of the above documentation should be made available to the participants of the DSARC or Service level reviews at least 10 days prior to the scheduled meetings. Where all data is not available (such as at an initial DSARC I), separate justification as to why it is not available should be presented. I have requested that OASD (I&L) act as a focal point for this second action and you will be hearing further from him.

By receipt of this memorandum it is requested that within 10 days the Services, OASD (I&L), and the chairman of the DoD Weapon System Software Steering Group reply by separate memorandum describing how they intend to comply with the actions initiated by this memorandum.

ATTACHMENT

FURTHER DESCRIPTION OF SUPPORTING DOCUMENTATION

1. Software Acquisition Management Plan

A software acquisition management plan should be prepared which as a minimum:

a. Identifies all major software elements (e.g., operational software, support software, automatic test equipment software, training software, simulation and validation software).

b. Describes the major life cycle phases of the weapon system and how the design, development, and maintenance of all software elements relate to them (specific periodic software management milestones should be identified).

c. Describes the software acquisition and procurement strategy to be followed.

d. Identifies major software related facility requirements over all life cycle phases (e.g., integration and validation facility).

e. Identifies specific software operational and maintenance resource requirements and describes how they will be addressed during early development phases.

f. Describes the management controls that will be followed.

g. Describes specific organizational roles and responsibilities.

h. Provide estimates of software related costs over all life cycle phases.

2. Analysis of New Software Developments and Facilities Versus Use of Existing Resources

A report should be prepared on the results of studies on the tradeoffs of using existing computer hardware, computer software, and/or facilities versus development of new computer hardware, computer software, and/or facilities. It is recognized that specific application programs will be different but it should be shown why an existing architecture within which the required application programs can function will not satisfy system requirements. Also it should be shown why existing support software, validation, test and evaluation, and maintenance software cannot be used.

3. Analysis of the Software Development Risks

A report should be prepared on the assessment of the risks involved with the development of software for the system. The operational software architecture and new or unique software that needs to be developed will be discussed along with difficulties that may be encountered due to mission and requirements uncertainties, software size, involvement of many organizations, contractor risks, georgaphically separated facilities, etc. The risk should be assessed in terms of cost, schedule, mission performance, reliability, and maintainability.

4. An Analysis of the Proposed Software Design Approach Versus Use of Hardware or Other Design Approaches

A report should be prepared presenting the results of a study on hardware/software tradeoffs for satisfying the required system capabilities with justification for each choice made.

APPENDIX H

FORMAT AND CONTENT
SAMPLE MATERIAL FOR USE IN PREPARING
WEAPON SYSTEMS SOFTWARE GUIDELINES

(Portions of general wording and outline
taken from existing 5000 Directives where possible)

1. Purpose

This Directive establishes policy for the acquisition of software (computer programs) in Department of Defense weapon systems. Emphasis of this Directive is on identifying specific software planning factors which should be considered in the preparation of the Decision Coordinating Paper (DCP) and in the Defense Systems Acquisition Review Council (DSARC) review process.

2. Applicability and Scope

The provisions of this Directive apply to the Office of the Secretary of Defense, the Military Departments, the Organization of the Joint Chiefs of Staff, and the Defense Agencies (hereinafter referred to collectively as "DoD Components") and encompass major weapon systems acquisition policies and programs. In addition, the acquisition principles in this Directive are applicable to non-major weapon systems and to major modifications to existing deployed weapon systems.

3. General

The DCP/DSARC process involves decision-making at the Secretary of Defense level on major defense system acquisition programs and related policies. The DCP and its supporting material document the current or proposed program and serve as the basis for DSARC reviews. The DSARC, as an advisory body, makes recommendations to the Secretary of Defense which are considered in the formulation of his decisions. The success of the DCP/DSARC process is vitally dependent upon a clear recognition of the individuality of each major defense system program and the sensible application of the policies of all the DoD 5000 series Directives and Instructions.

The policies described in this Directive are intended to supplement those in other 5000 series Directives and Instructions in areas where the unique characteristics of software require special OSD-level management attention.

4. General Weapon System Software Acquisition Policy

This section in final form would include specific OSD level software acquisition policies in each of the following areas (examples of possible wording and level of detail are included):

a. Software Requirements Formulation and Life Cycle Planning -- To insure consistent and comprehensive software requirements definition and total software life cycle planning in future DoD weapon systems, a software acquisition management plan will be required as DCP supporting material at the time of the initial DSARC I decision point and periodically updated thereafter. The minimum content and the procedures for the distribution and review of this plan are (to be) included as a separate enclosure to this Directive.

b. Control of Software Requirements During the Program -- To protect against software-related cost and schedule growth and computer hardware and software performance degradation caused by uncontrolled changes and user requirements growth, a system for prioritizing software requirements in major defense systems will be established. At the time of the initial software life cycle planning and requirements definition, software requirements will be identified as either high priority (essential to mission success); medium priority (necessary for most effective operation); or low priority (aids or nice features but not necessary for system operation). The requirements should be agreed upon with the user command prior to the start of the full-scale engineering phase and used to delete requirements when necessary to control cost, performance, and schedules during the program.

c. Evaluation of Use of Software Versus Hardware or Other Design Approaches -- To insure the use of software design approaches in weapon systems only when software represents the most beneficial design choice, a separate analysis of software versus other design approaches (including hardware, firmware, or manual procedures) will be performed during the initial validation phase. The analysis will be documented and provided as DCP supporting material at the time of the DSARC II decision point.

d. Consideration of Changes to Mission Requirements to Utilize Existing Computer Hardware, Software, and Related Facilities -- An analysis will be performed to identify computer hardware and software based subsystems and related facilities already in the DoD inventory with similar mission characteristics. The results of this analysis will be used at the time of the program initiation decision to determine if mission requirements should be adjusted to make use of all or portions of existing computer hardware, software, and facilities. It is important that this analysis be performed before software requirements are formalized (that is, before entering the validation/ contract definition phase). The results of the analysis will be documented and provided as DCP supporting material at the time of the DSARC I decision point.

e. Evaluation of New Software Developments and Facilities Versus Use of Existing Resources -- To insure the efficient utilization of existing DoD computer hardware, software, and facilities before initiating new developments, analysis will be performed during the initial validation phase. The analysis will consider use of existing computer hardware designs and software (including operating systems, application programs, support software; i.e., utility programs, languages, compilers, assemblers, testware, maintenance tools), and possible shared use of existing software maintenance and validation facilities. The analysis will be documented and provided as DCP supporting material at the time of the DSARC II decision point.

f. Analysis of Software Development Risks Involved -- To insure that significant software development risks are identified and that appropriate risk management methods are applied, a separate software risk analysis will be performed prior to the start of the validation phase and periodically updated thereafter. The analysis will be documented and provided as DCP supporting material at the time of the initial DSARC I decision point and updated for subsequent decision points.

g. Software Acquisition and Procurement Strategy -- When a significant amount of software development is (or is expected to be) involved in a weapon system, a specific software acquisition and procurement strategy should be developed at the time of the initial DSARC I decision point. This strategy should consider the software development risks involved, methods of providing contractor incentives, dependencies between software and other major subsystems, and overall schedules and methods for expediting them.

h. Software Prototyping -- Extensive use should be made of software prototyping when software development risks exist or user requirements are uncertain. In some cases, use of existing resources in existing systems should be used to demonstrate user requirements or simulate performance before entering into costly long-term software developments.

i. Monitoring and Validation of Software Development Activities -- An identifiable resource should be assigned to monitor and validate the activities of the software developer when significant amounts of software are involved. Use of in-house laboratory software personnel or a separate software validation contractor to supplement the project office is encouraged.

j. Software Integration and Test and Evaluation Facility -- A software integration and test and evaluation facility should be planned and available for software integration testing early in the software development cycle. Special software and hardware required to develop this facility should be included in the initial contract arrangements. Where feasible, facilities and personnel should be shared between weapon systems.

k. Software Interoperability Considerations -- To insure that adequate management attention and resources are applied to develop inter-system interface standards, to establish configuration management methods to control them, and to develop test methods for validating them, a separate software interoperability plan will be developed prior to the start of the validation phase and periodically updated thereafter. The plan will be provided as DCP supporting material at the time of the initial DSARC I decision point and updated for subsequent decision points.

l. Software Cost, Schedule, and Performance Information and Thresholds -- Separate software life cycle cost estimates, major software milestones (schedules), and software performance information will be provided in the DCP at each DSARC decision point. In addition, specific software cost, schedule, and performance thresholds will be agreed upon between OSD and the Service Components at each DSARC decision point. A definition of the software cost, schedule, and performance information to be provided are (to be) included as a separate enclosure to this Directive.

m. Exceptions in Decision Points Due to System Characteristics -- For some weapon systems, the DSARC decision points will align to the weapon platform milestones rather than to software (e.g., in an aircraft or missile). In such cases, software other than that needed to 'fly' the platform may be deemphasized during the initial validation phase (fly off). In these cases, separate intermediate software DCP/DSARC reviews and decision points are required.

5. Software Policy Relationships to Schedule Program Decision Points

This section in final form will include checklists of specific OSD-level software concerns to be emphasized at each DSARC decision point (examples of possible wording and level of detail are included).

Approval (or disapproval) to conduct a phase of a major defense system program will be given by the Secretary of Defense. The decision points shall be scheduled to meet the peculiar needs of each program. Each decision point shall be supported by a "for coordination" draft of a DCP and supporting material and a recommendation by the DSARC. The number, timing, and nature of the decision points shall be established by the Military Services and the Office of the Secretary of Defense (OSD) jointly and, though not the same for all programs, they will normally include:

a. DSARC I -- The Program Initiation Decision Point. At this decision point the Secretary of Defense considers approval (or disapproval) to commit resources for advanced development during the Validation Phase of a major defense system projected for inclusion in the force structure. Early scheduling of the program initiation decision point is essential to timely Secretary of Defense review. The primary software concerns at this decision point should include (in addition to those listed in other 5000-series Directives):

1. That all majcr (expected) software elements are identified.

2. That all software life cycle phases and phasing of all major software elements are identified.

3. That the overall software acquisition and procurement strategy is described.

4. That software prototypes, if applicable, are identified.

5. That the method for early validation of user software related requirements is identified.

6. That special software facility requirements are identified.

7. That a plan for development of software maintenance resources is identified.

8. That special software management controls are described.

9. That major software and user organizational roles and responsibilities are identified.

10. That major software development risks are identified and a plan/alternatives for removing the risks is available.

11. That a plan for evaluation of software design approaches versus other design approaches is described.

12. That existing computer hardware and software subsystems and related facilities with similar mission requirements have been considered for use rather than initiate new weapon systems or facility developments (that is, justification provided as to why realistic compromises in mission requirements cannot be made before program initiation to make use of all or portions of existing computer based weapon systems or facilities rather than to develop similar new ones).

13. That an ongoing plan for continued evaluation of existing resources and facilities versus new software developments and facilities is described.

14. That a plan for interface standardization to insure interoperability with other systems is described.

15. That all software life cycle cost estimates are provided.

16. That procedures for collection and reporting of software cost, schedule, and performance information have been defined and that specific DCP software thresholds for the Validation Phase have been agreed upon.

b. DSARC II -- The Full-Scale Engineering Development Decision Point. At this decision point, the Secretary of Defense considers approval (or disapproval) to commit resources to the full-scale engineering development or to the detailed design of a major defense system. The primary software concerns at this decision point should include (in addition to those listed in other 5000-series Directives):

1. That each item consider at the time of the DSARC I decision point (a. above) is updated with the results of activities conducted during the Validation Phase. Special emphasis at this time should be given to the following items.

2. That maximum utilization will be made of existing DoD computer hardware and software resources.

3. That maximum utilization will be made of existing DoD software facilities and in-house software personnel resources.

4. That the proposed software designs represent the best system approach having analyzed the cost, schedule, and performance tradeoffs of other alternatives (software versus firmware, hardware, or manual procedures).

5. That all software integration and validation facility requirements have been identified and planned for.

6. That the revised software acquisition and procurement strategy is consistent with the remaining software development risks.

7. That all software requirements are specified, priorities assigned, and agreed to with user command(s).

8. That a plan for an early software interoperability demonstration is provided.

9. That a software operations and support plan which identifies all required services and facilities required to maintain the software after deployment is available and that a plan for developing these services and facilities during the full-scale engineering phase is provided.

10. That the software cost, schedule, and performance information and corresponding DCP thresholds have been updated for the full-scale engineering phase and agreed upon.

c. DSARC III -- The Production/Deployment Decision Point. At this decision point, Secretary of Defense considers approval (or disapproval) to commit substantial resources to the production of a major defense system. The primary software concerns at this decision point should include (in addition to those listed in other 5000-series Directives):

1. That the performance of all software elements has been successfully demonstrated.

2. That all software operations and maintenance services and unique equipment and support tools are available for deployment.

3. That adequate computer hardware and software growth margins are available for field use.

4. That a software field deployment and integration plan is available.

5. That software interoperability has been demonstrated with all interfacing systems.

6. That all software requirement changes identified during full-scale engineering development are specified, priorities assigned, and agreed to with user command(s).

7. That adequate software documentation and configuration management procedures exist for field use.

8. That the software cost schedule, and performance information and corresponding DCP thresholds have been updated for the production/deployment phase and agreed upon.

d. Additional Decision Points. In addition to the three major decision points, the program situation may require additional decision points (e.g., completion of software prototype phase, limited production release, instances where major software decisions do not align with that of the weapon system platform, additional systems or facilities for test and evaluation, successive production lot procurement).

e. Unscheduled Program Decisions. Events both internal and external to the program (such as a congressional fund action, Secretary of Defense decision on a Program/Budget Decision, or a change in threat or national strategy), unforeseen technical difficulty or other circumstances -- which preclude achievement of a program objective or otherwise causes a breach, or a likely breach, of established cost, performance or schedule DCP thresholds -- may require a DSARC review in addition to those normally scheduled. Such reviews would lead to unscheduled program decisions.

6. The DCP/DSARC Process and the Program Memorandum (PM)

The PM is essentially the same as the DCP, but is used for programs which though important may not fully meet the criteria of DoD Directive 5000.1 as a major program warranting a DCP. The use of a PM to support program reviews and decision making shall be the same as the DCP except that (a) signature for approval shall be that of the appropriate Chairman of DSARC or at his discretion forwarded to the Secretary of Defense for signature; (b) the use of the DSARC to review the program shall be at the discretion of the DSARC Chairman; and (c) coordination on a PM may require that of the DSARC Chairman, Head of the DoD Component concerned, and only others having direct interest.

7. Waivers

Specific program circumstances may dictate the need for DoD Components to deviate from the policies outlined in this Directive (for weapon programs without significant software elements). When appropriate, the Head of the cognizant DoD Component may request a waiver to particular requirements of this document from the appropriate DSARC Chairman, indicating the circumstances that justify such waiver.

8. Responsibilities (Not Provided)

9. Effective Date and Implementation

Each DoD Component will implement this Directive within 90 days and forward copies of each implementing document to the Secretary of Defense.

Two Enclosures (Not Provided):

Content of Software Acquisition Management Plan

Definition of Software Cost, Schedule, Performance Information Requirements

APPENDIX I

DISTRIBUTION LIST

MITRE/Washington

H. D. Benington
C. C. Grandy
C. A. Zraket
Washington Library

D-40

H. J. Kirshner
D. L. Bailey
J. P. Locher III (25)
D. W. Kelliher (25)
A. Asch (5)
R. F. Robinson
W. Potter
L. G. Culhane
F. C. Holland
J. Greenwood

MITRE/Washington

G. Raichelson
J. Summers
K. Rebibo
E. Sharp
T. Hilinski
J. K. Summers
R. P. Foreman
E. L. Rabben

MITRE/Bedford

W. Attridge
J. Clapp
T. Connors (5)
O. Kinney
E. Lafferty
N. Waks
R. R. Everett

DoD

Col. R. D. Hensley (30)